

**UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



***SISTEMA DE LOCALIZACIÓN BASADO EN GPS Y
RECONOCIMIENTO VISUAL DE IMÁGENES***

**PROYECTO FIN DE CARRERA
INGENIERÍA SUPERIOR DE TELECOMUNICACIÓN**

**Autora: Ana Rodríguez Ordóñez
Tutor: Julio Villena Román**

Abril 2009

Título: SISTEMA DE LOCALIZACIÓN BASADO EN GPS Y RECONOCIMIENTO VISUAL DE IMÁGENES

Autor: Ana Rodríguez Ordóñez

Tutor: Julio Villena Román

EL TRIBUNAL

Presidente:

Manuel Urueña Pascual

Secretario:

Pedro Muñoz Merino

Vocal:

Alejandra Rodríguez Paniagua

Realizado el acto de defensa del Proyecto Fin de Carrera el día 27 de Abril de 2009 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo: Presidente

Fdo: Secretario

Fdo: Vocal

AGRADECIMIENTOS

Quiero dar las gracias a todas aquellas personas que, de un modo u otro, han sido parte de este proyecto.

En primer lugar, agradecer a mi tutor, Julio Villena, la confianza depositada en mí desde el primer momento. Gracias por su dedicación y compromiso con este proyecto, pues sin su ayuda no hubiese podido llevarlo a cabo.

Gracias a mi familia, por su empuje incondicional a lo largo de todos estos años de estudio. En especial, a mis padres, por haber estado ahí siempre que me ha hecho falta.

Gracias a Luis por su ayuda en los momentos difíciles, por su constante apoyo y por haberme animado siempre a seguir adelante.

Gracias a Belén, Ana, Marta Valdés y Marta Marinas por el cariño que me han dado y por las risas y los buenos momentos que hemos compartido.

Gracias a todos los compañeros de universidad, a Miriam, Virginia, Ana, Natalia...son sólo algunas de las personas que han hecho más fácil haber llegado hasta aquí.

Gracias a todos, porque una parte del mérito de este proyecto es también vuestra.

Este proyecto está dedicado especialmente a mi abuelo, a quien estoy segura le hubiese hecho mucha ilusión ver cerrada esta etapa que termina con la presentación de este proyecto.

RESUMEN

Hoy en día, existen multitud de aplicaciones que se basan en el reconocimiento de imágenes, o de forma más específica, en el reconocimiento de imágenes basado en contenido. La importancia que ha adquirido este campo, así como su potencialidad de crecimiento, han labrado vías de investigación en diferentes direcciones. Un caso concreto es el desarrollo de sistemas que combinan el reconocimiento de imágenes y su posicionamiento mediante unas coordenadas.

Este Proyecto Fin de Carrera tiene como objetivo implementar un sistema de localización basado tanto en posicionamiento GPS como en reconocimiento visual de imágenes. Para ello, se ha realizado un estudio exhaustivo de las distintas tecnologías existentes en ambos aspectos del sistema. A continuación, se ha realizado el diseño genérico del mismo y se ha llevado a cabo la implementación de cada una de las partes que lo componen. Por último, se han realizado las pruebas necesarias para demostrar la validez del sistema implementado y evaluar su comportamiento.

ABSTRACT

Nowadays, there are many applications which are based on image recognition, more specifically, on content-based image retrieval. The great importance acquired by this field, as well as its growth potentiality, have defined research trends in different directions. A specific case is the development of systems that combine image recognition and its location by means of a pair of coordinates.

This Master Thesis is aimed at implementing a location system based both on GPS location and visual image recognition. For this purpose, the work done started with an exhaustive research on the different technologies available in these two areas related to the project. Next, its generic design was carried out, and also the implementation of each of the elements that the system is composed of. Lastly, the necessary tests for showing the validity of the system and assessing its performance were carried out.

CONTENIDO

CONTENIDO	10
ÍNDICE DE FIGURAS	12
ÍNDICE DE TABLAS	13
1. INTRODUCCIÓN	14
1.1. DESCRIPCIÓN GENERAL DEL SISTEMA VPS.....	15
1.2. ESTRUCTURA DEL DOCUMENTO	17
2. CONCEPTOS PREVIOS	19
2.1. SERVICIOS BASADOS EN LOCALIZACIÓN	19
2.1.1. MÉTODOS DE LOCALIZACIÓN.....	19
2.1.2. APLICACIONES.....	20
2.2. SISTEMA GPS	21
2.2.1. HISTORIA DEL SISTEMA GPS.....	21
2.2.2. ELEMENTOS DE UN SISTEMA GPS.....	22
2.2.3. FUNCIONAMIENTO.....	23
2.2.4. OTROS SISTEMAS: GLONASS, GALILEO, COMPASS, IRNSS	24
2.2.5. APLICACIONES Y LIMITACIONES	25
2.2.6. FUTURO Y EVOLUCIÓN	26
2.3. RECUPERACIÓN DE INFORMACIÓN.....	26
2.3.1. RECUPERACIÓN DE INFORMACIÓN BASADA EN CONTENIDO	28
2.3.1.1. CARACTERÍSTICAS PARA LA RECUPERACIÓN DE IMÁGENES	29
2.3.2. TECNOLOGÍAS DISPONIBLES	32
2.3.2.1. SISTEMAS YA EXISTENTES	32
2.3.2.2. COMPARATIVA DE TECNOLOGÍAS	36
2.3.3. PROBLEMAS EN CBIR.....	38
2.3.4. CBIR EN DISPOSITIVOS MÓVILES	39
2.4. APLICACIONES EN DISPOSITIVOS MÓVILES.....	41
3. ARQUITECTURA DEL SISTEMA	44
3.1. MODELO CLIENTE-SERVIDOR	44
3.1.1. CARACTERÍSTICAS DE CADA ELEMENTO	45
3.1.2. TIPOS DE SISTEMAS CLIENTE-SERVIDOR.....	46
3.1.3. VENTAJAS E INCONVENIENTES	49
3.2. ARQUITECTURA DEL SISTEMA.....	50
4. DISEÑO DEL SISTEMA.....	52
4.1. DECISIONES DE DISEÑO.....	58

4.2.	BASES DE DATOS EMPLEADAS.....	61
4.3.	USO DE CBIR EN EL SISTEMA VPS	63
4.3.1.	JUSTIFICACIÓN DE DISTANCIAS.....	64
4.3.2.	JUSTIFICACIÓN DEL USO DE “RELEVANCE FEEDBACK”	66
4.4.	CÁLCULO DE LA DISTANCIA ENTRE DOS UBICACIONES	67
4.5.	DESCRIPCIÓN DE LAS CONSULTAS POSIBLES EN VPS.....	68
5.	IMPLEMENTACIÓN.....	74
5.1.	PROGRAMAS PHP DEL SERVIDOR.....	75
5.2.	SCRIPTS DE INDEXACIÓN DE FIRE	76
5.3.	SCRIPTS DE ARRANQUE DE FIRE	77
5.4.	PÁGINA WEB DE LA INTERFAZ.....	77
5.5.	MANUAL DE INSTALACIÓN Y CONFIGURACIÓN.....	78
6.	VALIDACIÓN DEL SISTEMA	79
6.1.	PRUEBAS REALIZADAS.....	80
6.2.	RESULTADOS	81
6.2.1.	CONSULTA DE LA DESCRIPCIÓN DE UN LUGAR.....	81
6.2.2.	CONSULTA DE UN LISTADO DE LUGARES CERCANOS	83
6.3.	ANÁLISIS DE LOS RESULTADOS	84
7.	CONCLUSIONES Y TRABAJOS FUTUROS	86
7.1.	ESCALABILIDAD	87
7.2.	TRABAJOS FUTUROS.....	88
	REFERENCIAS	90
	ANEXO I: INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA VPS	94
	ANEXO II: RESULTADOS DETALLADOS	98

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de un sistema de recuperación de imágenes basado en contenido	29
Figura 2. Interfaz de usuario del sistema <i>QBIC</i>	32
Figura 3. Interfaz de usuario del sistema <i>FIRE</i>	34
Figura 4. Interfaz de usuario del sistema <i>GIFT</i>	35
Figura 5. Interfaz de usuario del sistema <i>SIMPLcity</i>	36
Figura 6. Posibilidades de interacción en un sistema <i>CBIR</i> para dispositivos móviles	40
Figura 7. Arquitectura de 2 capas	47
Figura 8. Arquitectura de 3 capas	48
Figura 9. Arquitectura de n capas	48
Figura 10. Estructura del sistema <i>VPS</i>	50
Figura 11. Correspondencia con el modelo cliente-servidor	51
Figura 12. Posibles usos de la aplicación <i>VPS</i>	52
Figura 13. Páginas de inicio y de consulta de una descripción y de sitios cercanos	54
Figura 14. Páginas con resultados de consulta de una descripción y de sitios cercanos	56
Figura 15. Fragmento del documento <i>XML</i>	59
Figura 16. Estructura definida por el documento <i>DTD</i>	60
Figura 17. Estructura del documento <i>DTD</i>	61
Figura 18. Fórmula de Haversine para el cálculo de la distancia	68
Figura 19. Diagrama de flujo para la consulta de una descripción	69
Figura 20. Página de consulta de la descripción de un monumento	69
Figura 21. Diagrama de flujo para la consulta de una descripción	71
Figura 22. Página de consulta de la descripción de un monumento	72
Figura 23. Relación entre los programas que intervienen en la implementación	74
Figura 24. Medida de la tasa de error en el sistema <i>VPS</i>	79
Figura 25. Relación entre tiempo de análisis y tamaño de la base de datos	84
Figura 26. Estructura de directorios y ficheros del sistema <i>VPS</i> diseñado	94

ÍNDICE DE TABLAS

Tabla 1. Resumen de propiedades de los sistemas <i>QBIC</i> , <i>FIRE</i> , <i>GIFT</i> y <i>SIMPLicity</i>	37
Tabla 2. Características del cliente y del servidor	46
Tabla 3. Resumen de características de las bases de datos manejadas	62
Tabla 4. Lista de distancias disponibles en <i>FIRE</i>	64
Tabla 5. Compatibilidad entre las distancias y características de <i>FIRE</i>	65
Tabla 6. Distancias empleadas con las funciones de extracción de características	66
Tabla 7. Resultados con base de datos “A” para descripción de monumento.....	81
Tabla 8. Resultados con base de datos “A” para descripción de monumento con posición <i>GPS</i>	82
Tabla 9. Resultados con base de datos “B” para descripción de monumento.....	82
Tabla 10. Resultados con base de datos “B” para descripción de monumento con posición <i>GPS</i>	82
Tabla 11. Resultados con base de datos “C” para descripción de monumento.....	83
Tabla 12. Resultados con base de datos “C” para descripción de monumento con posición <i>GPS</i>	83

1. INTRODUCCIÓN

En los últimos años, arrastrado por el desarrollo de las nuevas tecnologías y el creciente interés de los usuarios por las aplicaciones derivadas, se han producido avances significativos en un campo que hasta hace poco tiempo permanecía inexplorado, como es la recuperación de imágenes.

Las investigaciones realizadas en esta área han abierto un abanico de aplicaciones basadas en estas tecnologías, por lo que se han desarrollado multitud de sistemas que se basan tanto en información sobre la posición como en el reconocimiento de imágenes de una base de datos.

El objetivo de este Proyecto Fin de Carrera es emplear estas dos tecnologías para llevar a cabo el diseño e implementación de un sistema de localización denominado *VPS* (*“Visual Positioning System”*). Esta aplicación consistirá en un servicio que permite localizar imágenes de una base de datos (que en el escenario concreto de aplicación de este proyecto contendrá fotografías de distintos lugares que la asemejen a una guía turística) y refinar los resultados proporcionados por un módulo de reconocimiento de imágenes mediante unas coordenadas *GPS* asociadas.

Con ello, se pretende no sólo posicionar las fotografías, sino además lograr una referencia de hacia dónde está mirando un posible usuario en el momento de tomar la imagen.

Al mismo tiempo, con la realización de este proyecto, se verá cómo el hecho de conocer la posición de un usuario aporta información adicional que permite lograr resultados más precisos en la búsqueda de información.

Asimismo, el aprovechamiento de la información contenida en una imagen a través de un sistema de reconocimiento basado en contenido, en vez de limitarse a los datos asociados a la misma por los usuarios, permitirá evitar la ambigüedad y la imprecisión que caracteriza las consultas basadas en indexación textual.

1.1. DESCRIPCIÓN GENERAL DEL SISTEMA VPS

Como bien se ha comentado, el objetivo de este Proyecto Fin de Carrera es diseñar una aplicación, *VPS*, que funcione en el escenario concreto de uso de este proyecto como una guía turística gracias a la base de datos definida.

El trabajo llevado a cabo se ha centrado en el sistema servidor, y se ha complementado con un cliente sencillo vía web, quedando fuera del alcance de este proyecto la implementación de un cliente mediante un desarrollo para dispositivos móviles.

Por tanto, este sistema plantea dos requisitos fundamentales: se debe disponer tanto de una fotografía como de las coordenadas *GPS* asociadas a la misma y que permiten georreferenciarla. Con estas premisas, la aplicación *VPS* diseñada permitirá realizar dos tipos de consultas:

- Si el usuario toma una fotografía de un monumento, dicha fotografía puede ser reconocida dentro de la base de datos de la aplicación, ayudándose de la información sobre la posición *GPS* si se dispone de ella, y el usuario puede obtener una descripción textual del monumento que está visualizando.
- Con la información sobre la posición *GPS* que proporciona el dispositivo móvil, el usuario puede conseguir una relación de monumentos a visitar situados como máximo a una cierta distancia del punto donde se encuentra. En cualquier caso, esta funcionalidad sería complementaria a la consulta de una descripción, ya que no lleva a cabo reconocimiento de las imágenes disponibles en la base de datos.

Como se verá más adelante, el diseño del sistema consistirá en implementar un servicio, cuya interfaz de usuario será una página web desde la que se puedan introducir tanto los datos sobre la ubicación *GPS* como la fotografía realizada.

Asimismo, los resultados serán presentados al usuario a través de la misma página. Esto permitirá que la aplicación sea accesible a través de un navegador desde todo tipo de dispositivos, incluidos terminales móviles.

1.2. ESTRUCTURA DEL DOCUMENTO

El presente documento se estructura de la siguiente manera:

- Capítulo 2: *Conceptos previos*

En este capítulo se realizará un estudio de las distintas tecnologías relacionadas con la aplicación VPS, como los servicios basados en localización LBS, la recuperación de información o el sistema de posicionamiento GPS.

- Capítulo 3: *Arquitectura del sistema*

Se realizará una descripción de la arquitectura genérica en la que se basa el sistema diseñado y se verán los motivos por los que se ha optado por dicha arquitectura.

- Capítulo 4: *Diseño del sistema*

En este capítulo se justificarán las decisiones de diseño tomadas y se hará una descripción detallada de cómo funciona la aplicación para las dos posibles consultas del usuario.

- Capítulo 5: *Implementación*

En esta sección se explicará de un modo más exhaustivo cómo se han desarrollado los distintos bloques de los que consta el sistema VPS.

- Capítulo 6: *Validación del sistema*

En este capítulo se describirá cuál es la configuración previa para el uso del sistema, y basándose en dicha configuración cuáles han sido las pruebas realizadas y qué resultados se han obtenido.

- Capítulo 7: *Conclusiones y Trabajos futuros*

Por último, en este capítulo se analizarán las conclusiones alcanzadas en la realización de este proyecto. Además, se comentarán posibles líneas de trabajo que permitan continuar el trabajo realizado, así como la escalabilidad y la flexibilidad del sistema ante futuras modificaciones.

- *Anexos*

Estas secciones recogen información adicional que completa lo explicado en algunos capítulos. El Anexo I contiene el manual de instalación y configuración que debe emplearse para utilizar la aplicación *VPS*. El Anexo II contiene los resultados experimentales obtenidos y sobre los que se basan las conclusiones obtenidas sobre las prestaciones del sistema diseñado.

2. CONCEPTOS PREVIOS

2.1. SERVICIOS BASADOS EN LOCALIZACIÓN

Los servicios basados en localización (en inglés “*Location Based Services*” o *LBS*) son servicios accesibles desde dispositivos móviles a través de la red móvil y que proporcionan al usuario información a partir de la posición geográfica [24]. Estos servicios son una muestra de la convergencia que se está produciendo en el sector de las telecomunicaciones, pues admiten aplicaciones que van desde la localización de personas hasta publicidad dirigida a los usuarios en función de su posición.

El funcionamiento de los *LBS* guarda una importante relación con los *GIS* (“*Geographic Information System*”), pues tienen en común el manejo de datos con una referencia espacial. Sin embargo, los *GIS* se pueden ver como sistemas profesionales dirigidos a usuarios experimentados y con un amplio abanico de funcionalidades. Por el contrario, los *LBS* son servicios con más limitaciones y orientados a un uso no profesional.

2.1.1. MÉTODOS DE LOCALIZACIÓN

El primer paso en un *LBS* es determinar la ubicación del usuario, para lo cual existen diferentes métodos:

- *GPS* (“*Global Positioning System*”): se trata de un sistema de posicionamiento vía satélite, del que más adelante se verá una descripción detallada. Es la solución estándar y que mayor precisión proporciona (el margen de error es de 3m a 20m), pero puede resultar costosa para el usuario pues el dispositivo móvil que emplee debe incorporar un equipo de este tipo.
- Localización *GSM* basada en celdas: permite averiguar la posición en función de la estación *BTS* a la que está conectada un teléfono móvil y la celda en la que se sitúa. Es una opción más económica pero menos precisa (de 100m a 5km), pues el teléfono no siempre se conecta a la antena más cercana al usuario, sino a la

que mejor señal proporciona. En las áreas rurales, donde las celdas son mayores y la densidad de estaciones es menor, se consigue una precisión limitada.

- *NLBS* (“*Near LBS*”): se emplean tecnologías de alcance local, como *RFID*, *WLAN*, Bluetooth o infrarrojos para asociar el dispositivo a un punto cercano y obtener su posición. En este caso, el usuario se sirve del entorno que le rodea para conseguir la información; es por ello que tiene mayor aplicación en entornos cerrados o de área local.

2.1.2. APLICACIONES

Los sistemas basados en *LBS* tienen aplicación en muchos escenarios diferentes, todos ellos tienen en común el hecho de que, a partir de una posición conocida, se puede poner a disposición del usuario la información que más se adecúe a sus necesidades.

Algunas aplicaciones de los *LBS* son la navegación giro a giro, la localización de recursos en un mapa, publicidad móvil en función de la posición o recepción de alertas para aviso de atascos de tráfico.

Desde el punto de vista del proveedor de servicios, los *LBS* aportan un valor añadido puesto que se pueden definir servicios de notificación (publicidad, datos de facturación en aeropuertos, etc.) o de actuación (por ejemplo, pago automático en peajes).

Gran parte de las aplicaciones basadas en *LBS* combinan los métodos de localización estudiados con el uso de mensajería móvil, sobre todo en el campo de la publicidad y el marketing. Un clásico ejemplo es el envío de ofertas y descuentos a los usuarios que se encuentran cerca de los establecimientos sujetos a estas promociones.

Sin embargo, las aplicaciones *LBS* han generado polémica acerca de la privacidad de los usuarios. Por ello, en algunos países existe normativa reguladora que obliga al proveedor del servicio a obtener del usuario la autorización para ser localizado, aunque

al mismo tiempo existe un vacío legal dado que no todos los casos posibles son contemplados por dicha normativa.

A pesar de todo, los *LBS* se encuentran hoy en día en profundo desarrollo, por lo que poco a poco irán apareciendo nuevas vías de perfeccionamiento y nuevas áreas en las que aplicar este tipo de servicios.

2.2. SISTEMA GPS

Como se ha visto anteriormente, el sistema *GPS* es uno de los posibles métodos de posicionamientos empleados en servicios basados en localización. A nivel mundial es el sistema más popular, por lo que a continuación se detalla su funcionamiento.

2.2.1. HISTORIA DEL SISTEMA GPS

Las siglas *GPS* corresponden a Sistema de Posicionamiento Global (en inglés “*Global Positioning System*”); es un sistema global de navegación por satélite que permite determinar la ubicación, velocidad o dirección de un objeto, persona o vehículo con un elevado nivel de precisión [8].

Aunque la idea de un sistema de navegación existía desde antes de la segunda Guerra Mundial, fue en 1957 cuando, a partir del lanzamiento del satélite Sputnik I por la Unión Soviética, empezó a desarrollarse la tecnología en la que se basan los *GPS*.

Un grupo de científicos estadounidenses que estaba monitorizando las señales retransmitidas por el Sputnik descubrió cómo variaba la frecuencia de la señal transmitida como consecuencia del efecto Doppler según el satélite se acercaba o se alejaba de ellos. De este modo surgió el sistema *TRANSIT*, empleado por la Marina de EE.UU., que quedó operativo en 1964 y fue comercializado en 1967. Posteriormente se diseñó una constelación de satélites basados en relojes atómicos y que estaban sincronizados con base a una referencia de tiempo determinada. Más tarde, en 1973, la Fuerza Aérea y la Marina de EE.UU. desarrollaron el “*Navigation Technology Program*”, del que

resultó el sistema *NAVSTAR-GPS* (nombre que recibe el sistema de navegación actual), cuya constelación de satélites alcanzó total operatividad en 1995.

2.2.2. ELEMENTOS DE UN SISTEMA GPS

El sistema *GPS* actual está compuesto por los siguientes elementos:

➤ Segmento espacial

Este segmento está formado por los satélites en órbita, también llamados “*Vehículos Espaciales*” (en inglés “*Space Vehicles*”, SV). En el diseño original el sistema *GPS* disponía de 24 SV's, ocho en cada una de las tres órbitas circulares, pero posteriormente se modificó de tal forma que existen cuatro órbitas con seis satélites cada una. La forma en la que los satélites están distribuidos permite que siempre haya línea de visión con seis de ellos desde prácticamente cualquier punto de la superficie terrestre. Desde Marzo de 2008 la constelación *GPS* dispone de 31 satélites activos, donde los adicionales mejoran la precisión de los cálculos que realiza el receptor *GPS* ya que permiten tomar medidas redundantes.

Cada uno de los satélites *GPS* emite de forma constante señales electromagnéticas con información sobre su posición orbital, la hora exacta de emisión de las señales y la posición del resto de satélites. El sistema de coordenadas empleado es el Sistema Geodésico Mundial, con centro fijo en la Tierra.

➤ Segmento de control

La trayectoria que describen los satélites es monitorizada de forma constante por cinco estaciones terrestres controladas por la Fuerza Aérea de los EE.UU., junto con otras estaciones gestionadas por la Agencia Nacional de Inteligencia Geospacial. La información sobre la trayectoria es enviada a una estación principal, desde donde se actualizan las cinco estaciones terrestres. Dichas actualizaciones permiten sincronizar los relojes atómicos que se encuentran en cada satélite, y ajustar la órbita que debe describir el mismo. Debido a la dificultad que supone mover un satélite para ajustar su órbita y a la precisión que requiere el movimiento, antes de modificar la posición de un

satélite se deja fuera de funcionamiento para evitar que los receptores lo consideren al realizar sus cálculos. Una vez realizado el movimiento, el satélite vuelve a funcionar y a describir la órbita calculada.

➤ Segmento de usuario

Está formado por el receptor *GPS*. En general, los receptores están compuestos por una antena, ajustada a las frecuencias empleadas por los satélites, un procesador y un reloj altamente estable (generalmente un oscilador de cristal). También suelen contar con una pantalla donde se proporciona al usuario información sobre su situación y su velocidad. Frecuentemente, se describen los receptores por su número de canales, es decir, el número de satélites que pueden monitorizar de forma simultánea. Aunque al principio contaban con cuatro o cinco canales, actualmente pueden monitorizar hasta veinte.

2.2.3. FUNCIONAMIENTO

El receptor *GPS* conoce la situación de los satélites en base a las efemérides, que son una serie de parámetros orbitales que son transmitidos por los propios satélites. Dado que la distancia del satélite al receptor y la posición del satélite son conocidas, el receptor *GPS* se encuentra situado en la superficie de una esfera que tiene como centro el satélite y como radio la distancia desde éste hasta el receptor. Obteniendo información de dos satélites se sabe que el receptor se encuentra sobre la circunferencia resultante de la intersección de las dos esferas. Adquiriendo información de un tercer satélite (triangulación), se obtienen dos puntos de corte, uno de los cuales podrá descartarse por representar una posición ilógica [17].

Aunque la posición sea en principio conocida, es necesario tener en cuenta que el reloj del receptor no se encuentra sincronizado con los relojes atómicos de los satélites, por lo que los puntos determinados no son precisos. Si se obtiene información de un cuarto satélite, se consigue eliminar la imprecisión causada por la falta de sincronización de los relojes y se obtiene la posición exacta del receptor *GPS* (latitud, longitud y altitud).

Una vez que se ha determinado la ubicación del receptor, se puede expresar en un sistema de coordenadas dado por latitud y longitud, empleando el Sistema Geodésico Mundial o un sistema local específico de cada país.

Finalmente, se pueden comprobar los resultados obtenidos o realizar ajustes a partir de los datos proporcionados por otros sistemas de posicionamiento, como *GLONASS* o Galileo.

2.2.4. OTROS SISTEMAS: GLONASS, GALILEO, COMPASS, IRNSS

Aunque el *GPS* es el principal sistema de navegación global por satélite, existen otros que en la actualidad no operan a pleno rendimiento o que se encuentran aún en fase de desarrollo.

Desde el año 2007, el sistema *NAVSTAR-GPS* desarrollado por EE.UU. es el único sistema de navegación global por satélite que se encuentra totalmente operativo. El sistema soviético *GLONASS* tuvo funcionalidad total pero tras la desaparición de la Unión Soviética presenta problemas de cobertura y sólo está disponible parcialmente. La Federación Rusa pretende devolver al sistema total operatividad para el año 2010 con la ayuda de India, que está participando en el proyecto.

Como alternativas al sistema *GPS*, la Unión Europea está actualmente desarrollando el sistema Galileo, el cual se estima que estará operativo en el año 2012 y que quedará bajo control civil. Sin embargo, este sistema será compatible con el *GPS*, de forma que los receptores puedan combinar las señales de ambos sistemas para aumentar su precisión.

Otro futuro sistema de navegación global es el planteado por China, *Compass*, que consiste en una extensión del sistema regional de navegación del que disponen actualmente (*Bigdoudou* o *Big Dipper*). Además, en India está siendo desarrollado el *IRNSS*

(“*Indian Regional Navigation Satellite System*”). Se trata de un proyecto que fue aprobado en 2006 y que será finalizado en 2012.

2.2.5. APLICACIONES Y LIMITACIONES

Dentro de las posibles aplicaciones de los sistemas *GPS* hay que diferenciar entre las militares y las civiles.

Entre las aplicaciones militares cabe resaltar la navegación (para localización de objetivos o coordinación de los movimientos de las tropas), el seguimiento de objetivos, el guiado de proyectiles, las tareas de reconocimiento para diseño de mapas o la localización para servicios de salvamento y rescate.

La mayoría de las aplicaciones civiles están basadas en la ubicación exacta del receptor, su movimiento relativo y su desplazamiento en el tiempo. Con estos datos, es posible emplear un sistema *GPS* como un medio de navegación, para localización de llamadas de emergencia, localización de personas, guiado turístico o rastreo y recuperación de vehículos.

Sin embargo, a pesar de que el *GPS* es el sistema de navegación más sencillo y preciso disponible en la actualidad, presenta una serie de limitaciones. Entre ellas, es importante destacar el hecho de que las señales emitidas por los satélites no pueden atravesar montañas, edificios, túneles o superficies metálicas, por lo que pueden originarse pérdidas temporales de la señal.

Además, en el caso de usar el receptor como medio de navegación, es necesario que el mapeado del sistema esté actualizado con el fin de garantizar una ruta óptima entre dos puntos, aunque ésta se pueda ver afectada por factores externos como el tráfico, las obras o la climatología.

2.2.6. FUTURO Y EVOLUCIÓN

Hoy en día el sistema de navegación por satélite tiene un futuro prometedor, ya que estos dispositivos cada vez tienen más aplicaciones (la mayoría de carácter civil).

Probablemente, el uso más novedoso de los sistemas *GPS*, y que actualmente sigue siendo desarrollado, es su combinación con teléfonos móviles. Estos receptores suelen ser diseñados específicamente para los terminales, pudiendo estar integrados o ser módulos independientes que se sincronizan mediante tecnología Bluetooth, y proporcionan datos de posicionamiento que son interpretados por un programa de navegación existente en el dispositivo móvil.

Debido a la constante evolución del *GPS*, cada vez se consiguen reducir más sus dimensiones, de modo que incluso permiten rastrear animales y personas ya que pueden integrarse en collares o pulseras de pequeño tamaño.

Sin embargo, un uso masivo del *GPS* puede plantear dudas acerca de la falta de privacidad que se podría alcanzar, ya que es posible que la posición de una persona sea rastreada constantemente. Esto puede hacer que la información obtenida sea utilizada con fines ilícitos, aunque por el contrario sí puede ser de gran utilidad para la actividad desarrollada por la policía u otros órganos oficiales.

2.3. RECUPERACIÓN DE INFORMACIÓN

La recuperación de información (*IR*, en inglés “*Information Retrieval*”) es la ciencia de la búsqueda de información en documentos, búsqueda de los mismos documentos, la búsqueda de metadatos que describan documentos, o, también, la búsqueda en bases de datos, ya sea a través de internet, de intranet, para textos, imágenes, sonido o datos de cualesquiera características [1].

Existe un cierto solapamiento en el uso de los términos recuperación de datos, recuperación de documentos, recuperación de información y recuperación de texto, pero cada uno tiene su propia base teórica y tecnologías. La *IR* es una ciencia

interdisciplinar, de la que se ocupan disciplinas tan diversas como la psicología cognitiva, la arquitectura de la información, la lingüística, la informática o la documentación.

Los procesos de recuperación de información se inician con la consulta de un usuario en un sistema. Las consultas son peticiones formales de una determinada información que el usuario necesita, como por ejemplo la búsqueda de palabras en motores web de búsqueda. En *IR* una consulta no identifica necesariamente un único objeto dentro de la colección, sino que puede haber varios resultados que se ajusten a la petición realizada (aunque quizá con diferentes grados de relevancia).

Un objeto es una entidad que almacena o representa información en la base de datos, por lo que las peticiones del usuario se asocian a objetos almacenados en ella. Dependiendo de la aplicación que emplee el usuario, los objetos pueden ser documentos de texto, imágenes, vídeos, etc.

La mayoría de los sistemas de *IR* calculan un valor numérico que representa el grado en que un objeto se ajusta a la consulta realizada, lo que permite ordenarlos de acuerdo a dicho valor. Así, los objetos cuya puntuación es más elevada son los resultados que se presentan al usuario, pudiendo realizar luego más iteraciones para refinar la búsqueda.

El objetivo de un sistema *IR* debe ser recuperar información que sea relevante para el usuario, o más exactamente, recuperar todos los documentos que sean relevantes intentando recuperar el menor número posible de documentos no relevantes.

Desde el punto de vista del usuario, es necesario conocer su comportamiento y sus necesidades ya que esto afecta de un modo muy directo la organización y la forma de operar del sistema. Por otro lado, desde el punto de vista técnico, la clave está en construir índices eficientes y algoritmos que proporcionen los resultados adecuados. Sin embargo, habitualmente ésta no es una tarea sencilla, debido a la ambigüedad e imprecisión que puede existir en la consulta realizada.

2.3.1. RECUPERACIÓN DE INFORMACIÓN BASADA EN CONTENIDO

Los sistemas de consulta de imágenes mediante ejemplo (también conocidos como sistemas de recuperación de imágenes basados en contenido, en inglés “*Content-Based Image Retrieval*” o *CBIR*) son sistemas de búsqueda para la recuperación de imágenes de grandes bases de datos a partir de su contenido [7]. Estos sistemas emplean el contenido visual de las imágenes, como forma, textura o color, para representar e indexar la imagen de forma automática, en vez de emplear una descripción textual proporcionada por el usuario como ocurre en los sistemas de indexación manual (que requieren mucho tiempo y en los que la descripción textual puede no ajustarse a las propiedades reales de la imagen). De este modo, los sistemas *CBIR* necesitan disponer de un motor de búsqueda que permita seleccionar las características relevantes para la consulta.

Se puede decir por tanto que, en los sistemas manuales, la determinación de las características la realiza el usuario inicial que introduce la imagen en la base de datos, que puede no ser el usuario final que realice la búsqueda, y por lo tanto, puede no especificar una característica innata de la imagen que requiera ser buscada posteriormente y determine la clave del éxito en la consulta. En cambio, en los sistemas automáticos, el que introduce los términos de búsqueda (color, formas, etc.) es el usuario final, y a partir de las características deseadas, el sistema realizará una búsqueda de las imágenes, teniendo en cuenta los deseos del usuario final y no las opciones de búsqueda que haya proporcionado el usuario inicial.

En los sistemas *CBIR* típicos, los contenidos visuales de las imágenes de la base de datos son extraídos y descritos a partir de vectores multidimensionales de características, de forma que los vectores de características de las imágenes de la base de datos forman una base de datos de características. Para recuperar una imagen, el usuario proporciona al sistema imágenes de ejemplo que posteriormente son convertidas a la representación interna del sistema, que viene dada por los vectores de características.

Mediante el cálculo de la distancia o similitud entre los vectores de características de la imagen de consulta y los de las imágenes de la base de datos, se realiza un indexado a partir del cual se recupera la imagen. Además del indexado, algunos sistemas modernos incorporan retroalimentación por relevancia por parte del usuario. Esto modifica el proceso de recuperación de la imagen y facilita que los resultados de la consulta sean más significativos tanto perceptual como semánticamente.

En la figura 1 se representa un sistema típico de recuperación de imágenes basada en contenido [7].

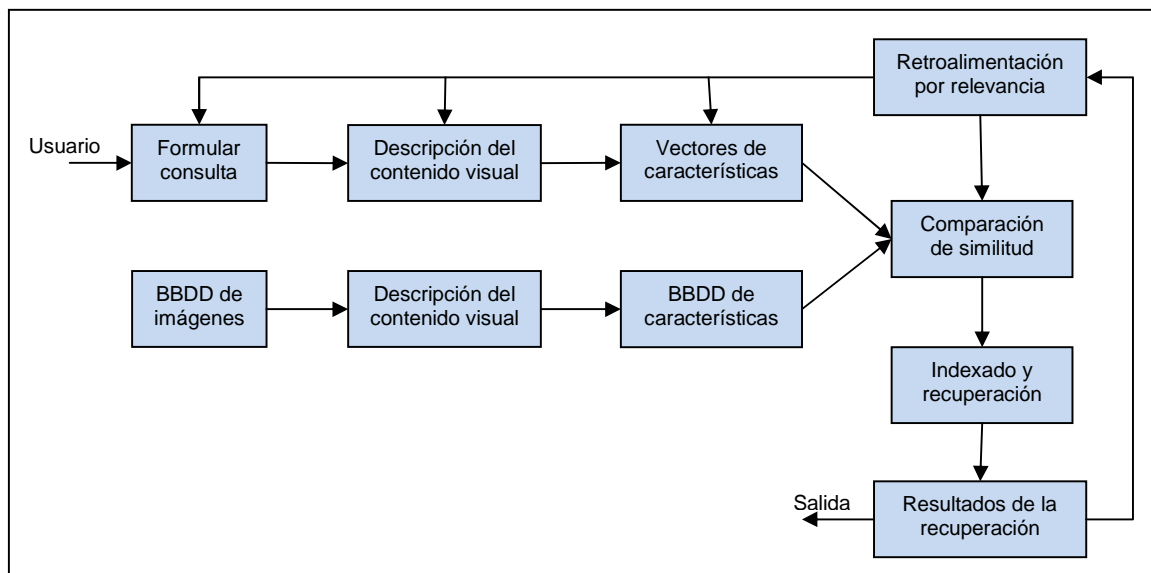


Figura 1. Arquitectura de un sistema de recuperación de imágenes basado en contenido

2.3.1.1. CARACTERÍSTICAS PARA LA RECUPERACIÓN DE IMÁGENES

Los sistemas *CBIR* realizan una comparación de imágenes basándose en unas ciertas características. Son muchas las características existentes, pero normalmente no se emplean todas porque esto supondría un volumen de datos y un tiempo de procesado muy elevados. Por tanto, se suele utilizar un subconjunto que sea adecuado para la tarea de recuperación de imágenes que se pretende llevar a cabo [5].

Dentro de los descriptores de contenido visual existentes, se pueden diferenciar dos tipos [2]:

- *Globales*: emplean las características visuales de la imagen completa (de cada píxel).
- *Locales*: para describir el contenido usan las características visuales de ciertas regiones (de cada píxel y una región próxima al mismo).

Mientras que las características globales son demasiado sensibles a la localización y pueden fallar en la identificación de ciertos contenidos visuales, el uso de características locales (junto con un paso añadido de unificación de características) permite incrementar la robustez ante transformaciones espaciales.

A continuación se describen algunas técnicas para la extracción de características:

➤ **HISTOGRAMA DE COLOR**

El histograma de color permite representar el contenido de color de una imagen siempre que el patrón de color sea único cuando se compara con el resto de la imagen. Esta técnica da una estimación de la distribución de los colores en la imagen. Para ello, el espacio de color (*RGB*, *CMY*, *HSV*, etc.) se divide y se calculan las frecuencias relativas de los colores que aparecen en una determinada partición de la imagen. Esta técnica es efectiva tanto en la caracterización global como local del color de una imagen.

Dado que el histograma de color no tiene en cuenta la distribución espacial, imágenes muy distintas pueden tener distribuciones de color similares. Esto se ve acentuado cuando el número de imágenes en la base de datos es muy elevado. Para aumentar la capacidad de discriminación de imágenes, se puede dividir la imagen en sub-áreas y calcular un histograma para cada una de ellas. Sin embargo, hay que tener en cuenta que, aunque esto aumenta la información relativa a la localización, también provoca un incremento del tiempo de procesado y de la memoria utilizada en el sistema.

➤ CARACTERÍSTICAS DE GABOR

Los filtros de Gabor son un tipo de filtro lineal que se usa con mucha frecuencia para el análisis de la textura de las imágenes. Esta técnica analiza la distribución de la intensidad en una imagen, lo que permite detectar la orientación de líneas y bordes presentes en la misma.

A partir de las características de Gabor de un conjunto de imágenes, es posible obtener histogramas y compararlos entre sí para obtener el resultado del sistema *CBIR*. Esto tiene gran utilidad en el reconocimiento de imágenes, como por ejemplo, del iris ocular o de las huellas dactilares.

➤ CARACTERÍSTICAS DE TEXTURA DE TAMURA

El conjunto de características de Tamura está formado por la granularidad, el contraste, la direccionalidad, el parecido de las líneas, la regularidad y la rugosidad. A partir de estudios de la percepción de la textura que tiene el ser humano, se determinó que las tres primeras características son las más relevantes, por lo que se podrían obtener histogramas a partir de ellas y comparar imágenes del sistema *CBIR*.

Sin embargo, cuando se busca distinguir texturas con mayor detalle, el uso de sólo tres características no es muy efectivo. Hace falta por tanto acudir a características basadas en estadísticas de señal de segundo orden, y que quedan definidas a partir de matrices de co-ocurrencia (definen la frecuencia de un par de niveles de color en dos píxeles separados una cierta distancia). Otra posibilidad sería usar un método basado en la *DCT* (en inglés, “*Discrete Cosine Transform*”), que tiene en cuenta las direcciones dominantes y las variaciones de color a lo largo de la imagen, y que por ello obtiene mejores resultados que la técnica de Tamura.

2.3.2. TECNOLOGÍAS DISPONIBLES

2.3.2.1. SISTEMAS YA EXISTENTES

A pesar de las limitaciones que tienen los sistemas *CBIR* disponibles actualmente, muchos se encuentran ya en el mercado a través de paquetes de software que incluyen demostraciones y versiones de prueba de los sistemas.

A continuación se describen algunos de los sistemas existentes.

➤ QBIC (Query By Image Content)

El sistema *QBIC*, que fue desarrollado por *IBM*, es uno de los sistemas *CBIR* más conocidos. Se trata de un sistema que permite realizar consultas a partir de cualquier combinación de palabras, colores, texturas o formas [9]. Para realizar la búsqueda de una imagen, se puede por ejemplo especificar una imagen de muestra o trazar una forma en la pantalla a partir de la cual se realiza la consulta. Comercialmente, el sistema *QBIC* está disponible de forma individual o como complemento de otros productos de *IBM*.

En la figura 2 se presenta la interfaz del sistema *QBIC*.



Figura 2. Interfaz de usuario del sistema *QBIC*

A partir de los datos introducidos, el sistema extrae las características de la imagen, las compara con las de las imágenes de la base de datos, y calcula una puntuación que permite ordenar los resultados de mayor a menor similitud y presentar por pantalla las imágenes más parecidas como resultado de la petición. También es posible combinar consultas basadas en el contenido con textos y palabras claves, para lograr una recuperación de imágenes más potente.

➤ **FIRE (Flexible Image Retrieval Engine)**

FIRE es un sistema de recuperación de imágenes basado en contenido que fue desarrollado por Thomas Deselaers en colaboración con la Universidad RWTH de Aachen. Ofrece un amplio repertorio de medidas de distancia y de características a partir de las cuales se pueden realizar consultas, lo que permite evaluar el funcionamiento de tareas de muy distinta índole [4]. Se trata, además, de un sistema de software libre.

En *FIRE*, al igual que en otros sistemas *CBIR*, las imágenes se obtienen a partir de sus propias características y no de palabras clave, por lo que al realizar una consulta se deben comparar las características asociadas a la imagen introducida con las de las imágenes de la colección. Dicha comparación se efectúa mediante funciones de similitud o distancias que son específicas para las diferentes características. Al mismo tiempo, pueden ser ponderadas mediante pesos según la importancia que se desee dar a cada característica.

En la figura 3 se presenta la interfaz de *FIRE*.



Figura 3. Interfaz de usuario del sistema *FIRE*

Además, *FIRE* permite emplear retroalimentación por relevancia (*“Relevance Feedback”*), una técnica ampliamente utilizada que refina los resultados de la búsqueda a partir de la interacción con el usuario: cuando se realiza una petición, el usuario determina qué subconjunto de los resultados obtenidos son relevantes, cuáles son irrelevantes y cuáles son neutros, y se hace una nueva petición a partir de los mismos, obteniéndose resultados acordes a la opinión del usuario.

➤ **GIFT (GNU Image Finding Tool)**

El sistema *GIFT*, desarrollado por el Departamento de Informática de la Universidad de Ginebra y de software libre, permite realizar consultas sobre imágenes mediante ejemplo, pudiendo además mejorar los resultados mediante el uso de retroalimentación por relevancia por parte del usuario [26].

Para procesar las peticiones, el sistema solamente tiene en cuenta el contenido de las imágenes, evitando que el usuario tenga que etiquetar todas las imágenes de la colección antes de realizar la consulta.

En la figura 4 se muestra la interfaz de usuario del sistema *GIFT*.

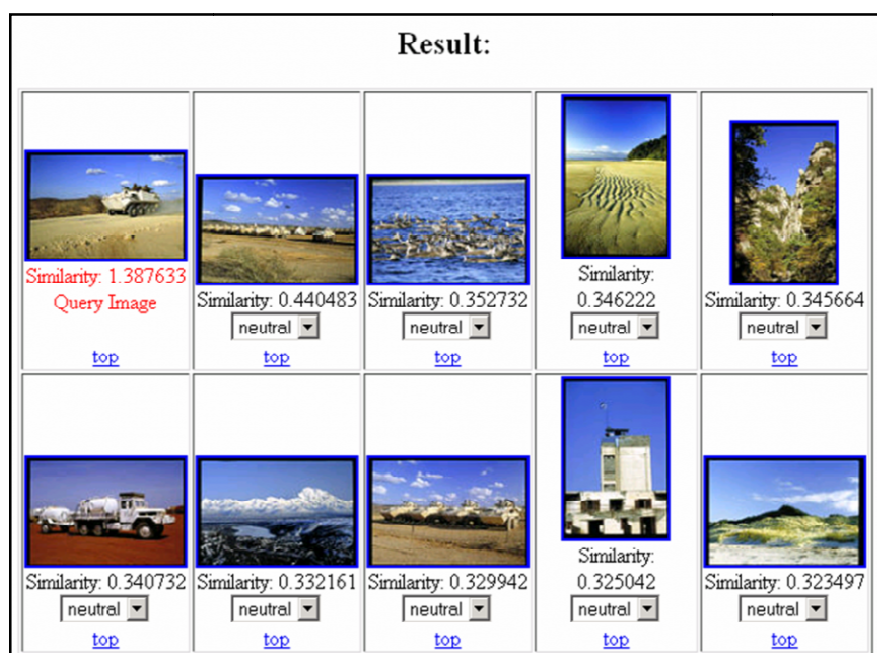


Figura 4. Interfaz de usuario del sistema *GIFT*

El software de *GIFT* permite indexar de una vez todas las imágenes contenidas en un determinado árbol de directorios. Esto posibilita que, empleando el cliente y el servidor de *GIFT* existentes, se localicen fotos en bases de datos de grandes dimensiones.

➤ **SIMPLicity (Semantics-Sensitive Integrated Machine for Picture Libraries)**

SIMPLicity es un sistema de recuperación de imágenes que emplea métodos de clasificación semántica y segmentación de la imagen para la identificación de la misma. Con este fin se divide la imagen en regiones que se caracterizan por su color, textura, forma y posición en la imagen global. Cada región de la imagen se cataloga dentro de una categoría semántica como “con textura”/“sin textura”, como “gráfico”/“fotografía”, etc. y se localiza la imagen estrechando la búsqueda a lo largo de las distintas categorías.

La figura 5 representa la interfaz gráfica de *SIMPLicity*.



Figura 5. Interfaz de usuario del sistema *SIMPLcity*

Para medir el parecido existente entre dos imágenes, se usa un esquema de identificación de regiones que combina las propiedades de todas las regiones de cada imagen. Comparada con la búsqueda basada en regiones individuales, este enfoque elimina fallos debidos a una mala segmentación de la imagen y concreta la semántica asociada a una región en particular.

2.3.2.2. COMPARATIVA DE TECNOLOGÍAS

De la descripción vista anteriormente para cada uno de los sistemas *CBIR* presentados, es posible deducir la tabla 1, en la que se resumen las distintas características que emplean cuando se realiza una petición y las distintas posibilidades:

Tabla 1. Resumen de propiedades de los sistemas *QBIC*, *FIRE*, *GIFT* y *SIMPLcity*

		SISTEMA			
		QBIC	FIRE	GIFT	SIMPLcity
CARACTERÍSTICA	Palabras clave	✓	×	×	×
	Colores	✓	✓	✓	✓
	Texturas	✓	✓	✓	✓
	Formas	✓	✓	✓	✓
	Distintas medidas o pesos para las características	×	✓	×	×
	Retroalimentación	×	✓	✓	×
	Indexado de directorios	×	×	✓	×
	Segmentación	×	×	×	✓
	Categorías semánticas	×	×	×	✓

Como se puede apreciar, el sistema *QBIC* es el más sencillo de todos, lo cual es lógico dado que es el sistema más antiguo de los presentados y menos evolucionado.

En cuanto al resto de sistemas, se observa que ninguno permite realizar consultas basadas en palabras clave y que todos utilizan características como los colores, las texturas o las formas presentes en la imagen consultada.

Por otro lado, se podría decir que cada sistema está orientado de forma distinta a tareas diferentes. Por ejemplo, *SIMPLcity* no permite interacción con el usuario mediante realimentación, posible en *FIRE* y *GIFT*, pero por el contrario permite clasificar la imagen semánticamente.

En general, *FIRE* es un sistema más completo que el resto. Se trata de un sistema muy parecido a *GIFT* pero, aunque a diferencia suya no permita indexar directorios de una sola vez, el hecho de que se puedan combinar las características mediante pesos según la tarea que se desee llevar a cabo lo convierte en un sistema más flexible y con más posibilidades.

2.3.3. PROBLEMAS EN CBIR

Aunque *CBIR* se ha desvelado en los últimos años como una tecnología con grandes posibilidades, supone una serie de problemas que afectan a todos los sistemas de recuperación de imágenes, ya sea directa o indirectamente.

Existen dos limitaciones principales en los sistemas *CBIR* [23]; la primera está relacionada con los descriptores que se emplean y su capacidad para identificar imágenes en casos de variación de escala, rotación, traslación o iluminación, y la segunda con la velocidad de búsqueda en tiempo real, especialmente en escenarios de movilidad. Por tanto, los descriptores deben ser de tamaño limitado e invariantes a los cambios antes mencionados, así como adaptables a la totalidad de las imágenes de la base de datos.

Otro inconveniente que se deriva de la necesidad de entender cuál es realmente el contenido de una imagen es lo que se conoce como anotación automática, que puede simplificar la búsqueda de imágenes a través del empleo de texto [2]. Sin embargo, para conseguir esto es preciso que la asociación entre el texto y las imágenes sea fiable, y sólo en este caso será beneficioso incorporar texto en la consulta.

A este problema se une la dificultad de relacionar con la imagen una porción de texto en vez de palabras individuales, pues se añade el carácter subjetivo que tiene el texto para el usuario y el hecho de que en una base de datos habrá posiblemente más de una imagen que pueda representar las ideas expresadas en el texto.

A pesar de las complicaciones que se plantean, las investigaciones que se están realizando sobre estas cuestiones conducirán probablemente a nuevas oportunidades y vías de innovación que permitan desarrollar sistemas en los que estas limitaciones no estén presentes.

2.3.4. CBIR EN DISPOSITIVOS MÓVILES

A pesar de que ya se ha visto cuál es el funcionamiento de los sistemas *CBIR* y qué tecnologías relacionadas existen, es imprescindible profundizar en la importancia que ha adquirido esta técnica en el área de la tecnología móvil. Más adelante, se estudiarán en detalle algunas aplicaciones que hacen uso de *CBIR* y que son específicas para dispositivos móviles.

La proliferación del uso de *CBIR* en este campo se debe a la necesidad de poder realizar consultas en grandes bases de datos de imágenes y poder obtener resultados en función del contenido de una imagen, necesidad que se ha visto reforzada por los avances en tecnología y el aumento del número de plataformas móviles que soportan servicios multimedia.

Así como los *LBS* se han convertido en servicios muy populares hoy en día, se ha empezado recientemente a utilizar los sistemas *CBIR* en dispositivos móviles y probablemente la difusión de estos sistemas aumentará de cara al futuro. Cuando se emplea un dispositivo móvil, para el usuario resulta especialmente incómodo tener que realizar la indización manual de una imagen para proceder a una consulta sobre la misma, por lo que el uso de *CBIR* puede simplificar en gran medida este tipo de consultas. Es por ello que actualmente se están llevando a cabo multitud de investigaciones, las cuales permitirán mejorar los sistemas de consulta mediante ejemplo aplicados a plataformas móviles.

Dentro de las aplicaciones basadas en *CBIR* y desarrolladas para terminales móviles, se pueden distinguir dos tipos: las que están integradas en el terminal, y las que se sirven de tecnología *SMS*, *MMS*, o *WAP* para que el usuario haga una consulta a un servidor donde esté almacenada la información.

Como se ha visto anteriormente, muchos de los servicios empleados en este campo se ayudan de la información sobre la posición del usuario para proporcionar resultados. Es por ello que algunas aplicaciones disponibles se pueden considerar como una combinación de sistema *CBIR* y servicio *LBS*. La información acerca de la ubicación puede ser obtenida por *GPS*, localización basada en celdas u otros métodos, pero, en

cualquier caso, si está acompañada de información más explícita (como es una imagen) podrá emplearse una tecnología *CBIR* y los resultados de la petición serán más satisfactorios.

Estas distintas modalidades de consulta pueden clasificarse de la manera mostrada en la figura 6.

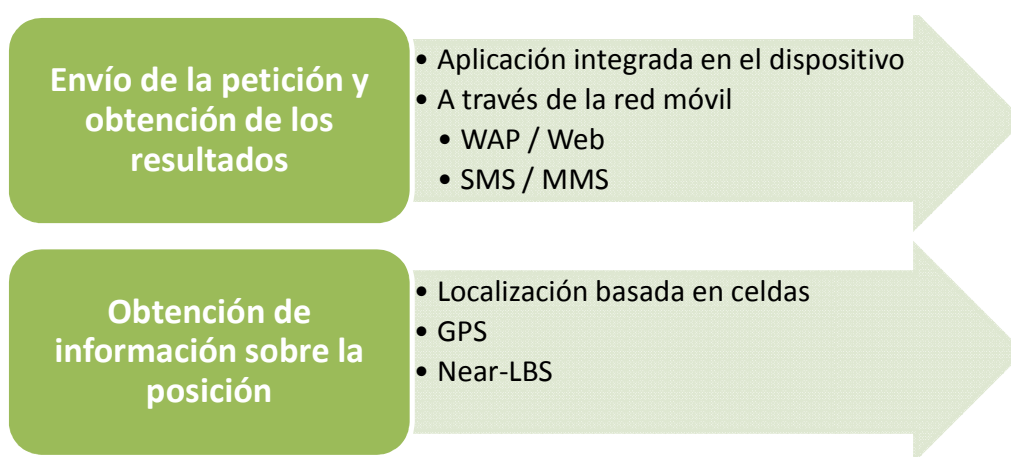


Figura 6. Posibilidades de interacción en un sistema *CBIR* para dispositivos móviles

Las diferentes formas de realizar la consulta y obtener el resultado, así como el hecho de poder obtener opcionalmente información sobre la ubicación del usuario, determinan un amplio abanico de posibles aplicaciones en el campo de la recuperación basada en imágenes, como se verá más adelante en este trabajo.

En resumen, se puede decir que la aplicación de *CBIR* a los dispositivos móviles sirve de nexo entre el mundo físico que rodea al usuario y el mundo virtual, puesto que además la introducción de información por parte del usuario a través de este tipo de terminales requiere muy poco tiempo y aumenta la usabilidad de los sistemas.

2.4. APLICACIONES EN DISPOSITIVOS MÓVILES

Gracias a las amplias posibilidades que ofrecen tanto los servicios basados en localización como la recuperación de imágenes basada en contenido, han aparecido en el mercado multitud de aplicaciones, muchas de las cuales son adecuadas para el uso en dispositivos móviles, como las que se verán a continuación.

Dichas aplicaciones pueden diferenciarse en función de si utilizan reconocimiento de imágenes tomadas por el usuario, la posición correspondiente o ambas cosas.

➤ RECONOCIMIENTO DE IMÁGENES

Existen multitud de aplicaciones basadas en reconocimiento de imágenes, que van desde la identificación de códigos bidimensionales hasta el reconocimiento de carátulas de discos de música.

Los códigos bidimensionales son imágenes relativamente sencillas de reconocer, ya que se trata de una matriz de puntos en blanco y negro donde la información viene codificada según la forma y la orientación de dichos puntos. El usuario puede almacenar códigos en su teléfono y, empleando lectores electrónicos, se pueden identificar y obtener información de promociones publicitarias, datos de la tarjeta de embarque para billetes de avión, información sobre un producto localizada en su envase, etc.

Un sistema de mayor complejidad es el conocido como *PhoneGuide* [10], que permite realizar un recorrido por un museo identificando los objetos expuestos en él y obteniendo información sobre los mismos. Este sistema cuenta con la ventaja de poder realizar todo el proceso de reconocimiento en el propio dispositivo móvil, lo que evita que haya tráfico por la red móvil y hace que se reduzca el coste.

Entre las aplicaciones más completas que emplean reconocimiento de imágenes, cabe mencionar un sistema que sirve para identificar la posición de un dispositivo a partir de búsqueda de imágenes en la Web (pero sin tener un dispositivo con módulo *GPS* o de localización) [27]. El funcionamiento de este sistema es el siguiente: a partir de una

imagen tomada con el teléfono, se reconoce dicha imagen dentro de una pequeña base de datos donde las imágenes tienen asociadas palabras clave; luego se realiza una búsqueda textual en *Google* con las palabras clave asociadas a la imagen escogida, y se obtiene un conjunto de imágenes que es filtrado para obtener como resultado aquéllas cuya posición sea más cercana a la de la imagen original.

➤ POSICIÓN ASOCIADA

Este tipo de aplicaciones guarda una mayor relación con los servicios *LBS*, ya que el concepto de localización es la base de su funcionamiento.

Dentro de las aplicaciones comerciales, hay algunas cuyo objetivo es la localización de personas, como es el caso de *LocalizaGPS+*, una solución con la que se puede localizar dispositivos que cuenten con un módulo *GPS* y almacenar información sobre la posición en una base de datos. Esta aplicación permite la integración total con la cartografía de *GoogleEarth* y *GoogleMaps*, lo que facilita el seguimiento en tiempo real desde un ordenador.

De carácter muy diferente es el sistema *LocationWeb* [19], que permite crear y buscar contenido web basado en la posición desde un teléfono móvil. El teléfono emplea los datos sobre la ubicación como metadatos del contenido web y posibilita la búsqueda en tiempo real basada en información de la posición.

➤ RECONOCIMIENTO DE IMÁGENES Y POSICIÓN

Mediante la combinación de los dos tipos de aplicaciones descritos anteriormente, se han desarrollado algunos sistemas que, aunque tienen una estructura más compleja, tienen al mismo tiempo más posibilidades y mayor utilidad.

Por ejemplo, *SnapToTell* [16] es un sistema que proporciona información sobre servicios útiles para turistas basándose en fotografías tomadas con un dispositivo móvil y en datos sobre la localización. Para obtener información sobre un restaurante, un objeto expuesto en un museo o cualquier otro elemento turístico, el usuario realiza una foto con el teléfono móvil y lo envía mediante un mensaje multimedia a un proveedor de

servicios, el cual le enviará un mensaje de texto o multimedia con la información solicitada.

Otra aplicación que también emplea mensajes de un proveedor de servicios es el sistema *MIR* ("*Multimedia Information Request*"), que combina tecnología de la red móvil (*SMS*, *MMS* y *WAP*) con técnicas de recuperación de imágenes basada en contenido [23]. La finalidad de este servicio es facilitar la búsqueda de información sobre monumentos históricos y de interés cultural en Marruecos. Cuando un usuario realiza una foto, la puede enviar por *MMS* a un servidor donde se realiza la localización geográfica y se obtienen resultados en función de las similitudes con otras imágenes previamente almacenadas. Los resultados se pueden visualizar de dos formas, bien como texto detallado en un *SMS* que recibe el usuario, o bien a través de un portal *WAP*.

3. ARQUITECTURA DEL SISTEMA

La arquitectura en la que se basa el sistema *VPS* es de tipo cliente-servidor. El motivo por el que se ha decidido estructurar el sistema de este modo es que, como se verá más adelante, permite que haya varios usuarios utilizando la aplicación *VPS* y haciendo consultas al servidor de forma simultánea. Además, en el sistema diseñado los usuarios (que asumen el papel de clientes) sólo realizarán las peticiones, mientras que será el servidor el que maneje todos los datos.

A continuación se comenta más en detalle el funcionamiento de la arquitectura cliente-servidor, así como su relación con los módulos de los que consta el sistema implementado.

3.1. MODELO CLIENTE-SERVIDOR

Debido a los avances tecnológicos en sistemas de computación y a las necesidades y posibilidades demandadas por el mercado, hace ya tiempo que se hizo necesario abandonar el modelo de procesamiento centralizado (característico de los años 70), en el que un ordenador realiza todo el procesamiento, por el modelo de procesamiento distribuido, en el que varios ordenadores distribuidos físicamente y conectados por una red asumen todas las funciones.

Según la topología, se pueden distinguir dos tipos de arquitectura distribuida:

- **Arquitectura descentralizada:** todos los componentes realizan las mismas funciones.
- **Arquitectura centralizada:** cada componente asume un diferente papel.

Un caso de modelo descentralizado es la arquitectura de pares (“peer-to-peer”), donde cada elemento puede actuar al mismo tiempo como cliente y como servidor. Dentro de las arquitecturas centralizadas, se encuentran las de tipo multicapa (“n-tier”), en la que los componentes emiten peticiones y también responden a peticiones de otros

elementos que actúan como clientes. Un caso particular de arquitectura de este tipo es el modelo cliente-servidor, que ha sido el empleado en el diseño de este proyecto.

La arquitectura cliente-servidor es un modelo de interacción entre un programa cliente que realiza peticiones y un programa servidor que le proporciona respuestas. Se basa, por tanto, en el concepto de servicio implementado sobre un diálogo petición-respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre un solo ordenador, es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de ordenadores [28].

En este modelo la capacidad de procesamiento se reparte entre los clientes y los servidores, pero son más importantes los beneficios de tipo organizativo debidos a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

3.1.1. CARACTERÍSTICAS DE CADA ELEMENTO

En la arquitectura cliente-servidor, el remitente de una solicitud es conocido como cliente, mientras que el receptor de la solicitud se conoce como servidor. Sus características son las siguientes:

➤ Cliente

- Inicia solicitudes y peticiones, tiene por tanto un papel activo en la comunicación.
- Espera y recibe respuestas por parte del servidor.
- En general puede conectarse a varios servidores a la vez.
- Normalmente no comparte sus recursos con otros clientes.
- Suele interactuar directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- No tiene restricciones especiales respecto a rendimiento, fiabilidad y escalabilidad, por lo que no necesita un equipo de elevadas prestaciones, y el fallo de un cliente no afecta al resto del sistema.

➤ Servidor

- Espera solicitudes del cliente, asumiendo un papel pasivo en la comunicación.
- Procesa las solicitudes emitidas por el cliente, elabora las respuestas y las envía al cliente.
- Generalmente acepta conexiones de un gran número de clientes, aunque en algunos casos el número máximo de conexiones puede estar limitado.
- Normalmente comparte sus recursos con los clientes a los que da servicio.
- Raramente interactúa con los usuarios finales.
- Tiene restricciones de rendimiento, fiabilidad, escalabilidad y seguridad, por tanto requiere un equipo con capacidad suficiente para atender múltiples clientes, y un fallo en el servidor es crítico e invalida el sistema completo.

En la tabla 2 se resumen las características mencionadas anteriormente.

Tabla 2. Características del cliente y del servidor

		COMPONENTE	
		CLIENTE	SERVIDOR
CARACTERÍSTICA	Realiza peticiones	✓	x
	Emita respuestas	x	✓
	Conexión múltiple	✓	✓
	Comparte recursos	x	✓
	Interactúa con usuarios finales	✓	x
	Restricciones de rendimiento, fiabilidad, seguridad	x	✓

3.1.2. TIPOS DE SISTEMAS CLIENTE-SERVIDOR

La arquitectura cliente-servidor tiene dos tipos de nodos en la red: clientes y servidores. En función de la ubicación física de dichos nodos y de la funcionalidad que asume cada uno se diferencian varios tipos de sistemas cliente-servidor:

➤ Arquitectura de 2 capas

En este modelo el cliente realiza peticiones y el servidor responde a ellas directamente, utilizando sus propios recursos. Esto significa que el servidor no necesita otra aplicación para proporcionar parte del servicio. La figura 7 representa este modelo.

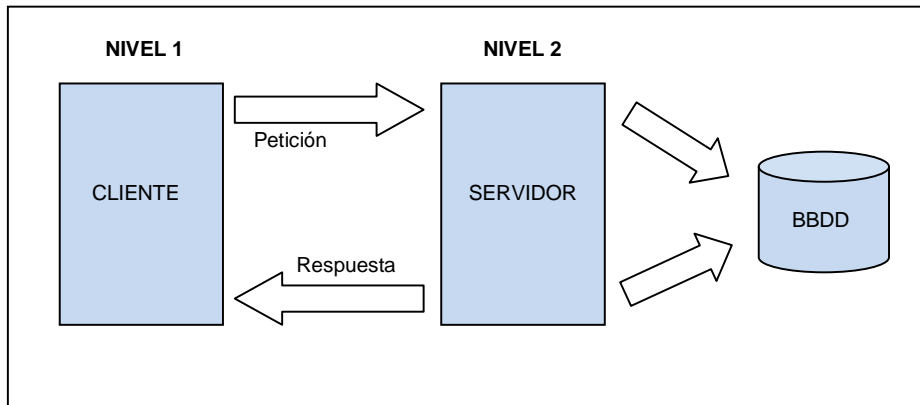


Figura 7. Arquitectura de 2 capas

Como se aprecia en la figura, existe un único servidor que atiende a los clientes, lo que genera un alto grado de dependencia con el servidor. Esto hace que se trate de un modelo poco escalable y demasiado rígido cuando se trata de sistemas complejos, ya que cualquier variación implica grandes cambios en la arquitectura y en la funcionalidad de cada elemento.

➤ Arquitectura de 3 capas

En este tipo de arquitectura existe un nivel intermediario, quedando el servidor dividido en dos componentes. Generalmente, se tratará de un servidor de aplicaciones que procesa los datos para los clientes, así como de un servidor de base de datos que proporciona los datos al servidor de aplicaciones. El esquema se muestra en la figura 8.

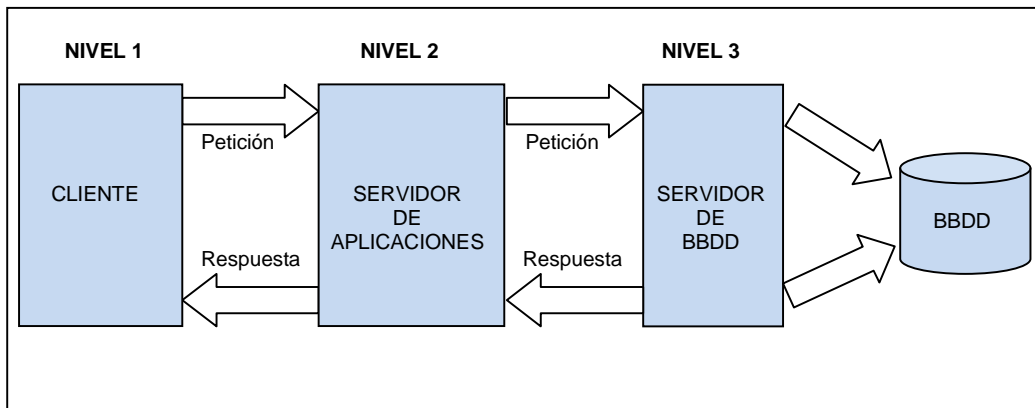


Figura 8. Arquitectura de 3 capas

➤ Arquitectura de n capas

Este tipo de arquitectura es una generalización del modelo de 3 capas. La lógica de aplicación se reparte en diferentes niveles ubicados entre el cliente y los datos, de manera que dichos niveles intermedios realizan tareas específicas y se proporcionan servicios entre sí a través de interfaces bien definidos. En la figura 9 se muestra esta arquitectura.

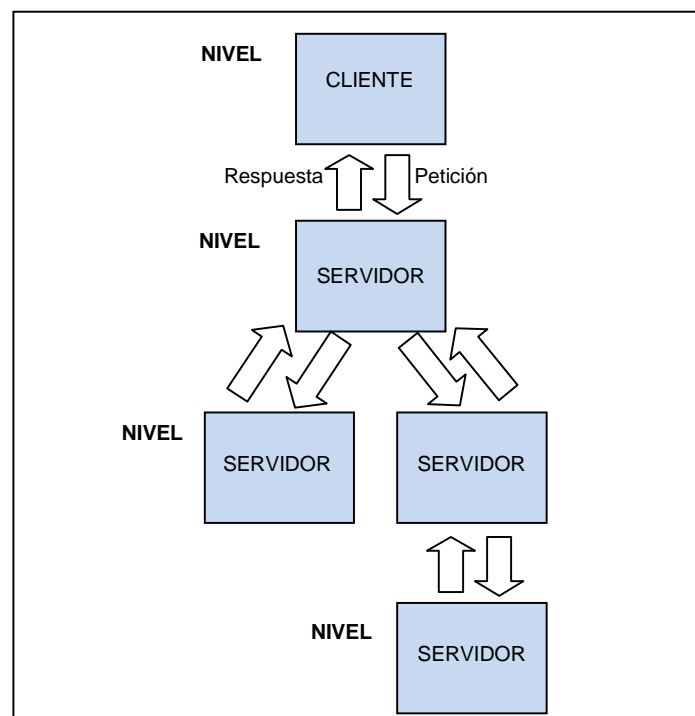


Figura 9. Arquitectura de n capas

La configuración en n niveles tiene la ventaja fundamental de que permite separar el proceso hacia fuera, balanceando la carga en los diversos servidores y dotando al sistema de mayor escalabilidad. Además, proporciona un elevado grado de seguridad, pues es sencillo propagar la autenticación a lo largo de las distintas capas y aislar unas capas frente a cambios que se produzcan en otras.

Sin embargo, es una arquitectura más compleja y por tanto más difícil de gestionar y mantener, lo que se traduce en costes mayores. Asimismo, hace que la carga en la red sea más elevada, pues entre los distintos niveles existe tráfico que no está presente en el modelo de dos capas.

3.1.3. VENTAJAS E INCONVENIENTES

Aunque los sistemas basados en arquitectura cliente-servidor tienen como objetivo hacer más accesibles los recursos disponibles y suponen multitud de ventajas, la configuración adoptada en dicho modelo implica también una serie de desventajas.

➤ Ventajas

- *Centralización del control:* los accesos, los recursos y la integridad de los datos son controlados por el servidor, lo que además facilita la tarea de mantenerlos al día.
- *Escalabilidad:* la capacidad de servidores y clientes puede ser aumentada por separado.
- *Fiabilidad:* se puede mejorar la tolerancia a fallos replicando los recursos críticos.
- *Transparencia:* el hecho de que los recursos estén distribuidos es transparente al usuario, que no percibe diferencias de ubicación, de concurrencia de uso, o de acceso a recursos.

➤ Inconvenientes

- *Robustez:* si el servidor no está disponible, es imposible satisfacer peticiones de ningún cliente, al contrario que en las redes de pares.
- *Desarrollo de software:* normalmente debe ser específico, sobre todo del lado del servidor, para poder satisfacer el trabajo, lo que supone costes más elevados.

- *Congestión de tráfico:* cuando una gran cantidad de clientes envían peticiones simultáneas al servidor, éstas no pueden ser satisfechas.

3.2. ARQUITECTURA DEL SISTEMA

El funcionamiento de la aplicación *VPS* se basa en un servicio web, en el que existen peticiones de información que hace un usuario a través de una página web y que obtienen como respuesta información contenida en la base de datos de la aplicación. Dicha información es proporcionada por un servidor y entregada al usuario a través de la página web.

En la figura 10 se muestra la estructura del sistema diseñado.

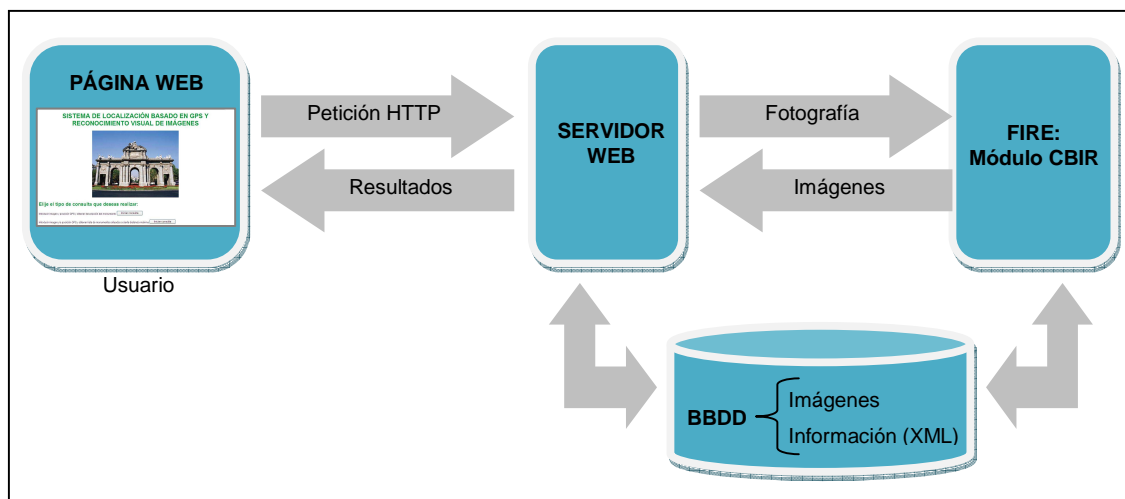


Figura 10. Estructura del sistema *VPS*

Cuando un usuario desea formular una consulta, accede a la página web que actúa como interfaz y selecciona el tipo de petición que desea realizar (obtener bien la descripción de un monumento o bien una lista de sitios cercanos). A continuación se le solicita que especifique los datos sobre los que se va a realizar la consulta, y estos datos son enviados al servidor mediante una petición *HTTP* (*"HyperText Transfer Protocol"*) que extrae la información del formulario rellenado por el usuario en la página.

Una vez se han enviado los datos al servidor, los siguientes pasos del proceso vienen determinados por el tipo de consulta realizada. En cualquier caso, el servidor localizará en la base de datos la información pertinente y la entregará al usuario mediante su presentación en la página web. Además, si el usuario ha aportado una fotografía, ésta será proporcionada al módulo *CBIR* de reconocimiento de imágenes (implementado en este proyecto mediante el sistema *FIRE*), que facilitará las imágenes más parecidas y serán también mostradas a través de la página web.

Como se puede observar, la página desde la que se envían las peticiones tiene el papel del cliente dentro de una estructura cliente-servidor. Por otro lado, el servidor web y el módulo de reconocimiento de imágenes, que son los que acceden a la información de la base de datos, asumen la función de servidor dentro de dicho modelo.

Además, podría haber varios clientes accediendo al mismo tiempo al servidor, en caso de que varios usuarios realicen consultas a través de la aplicación al mismo tiempo.

Dentro de los tipos de arquitectura cliente-servidor que se han explicado en este documento, el sistema *VPS* correspondería a un modelo de 2 capas, ya que sólo existen en este caso dos tipos de nodos: por un lado, el cliente, y por otro, el servidor que accede a los datos (el módulo *CBIR* está en el lado del servidor).

En la figura 11 se puede ver la analogía del sistema *VPS* con el modelo cliente-servidor.

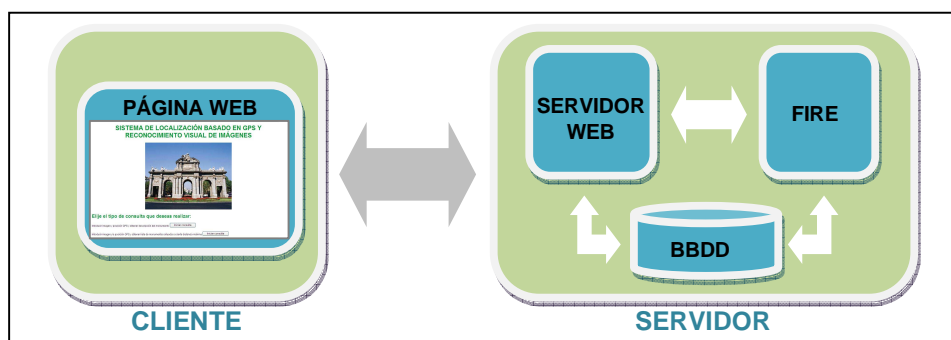


Figura 11. Correspondencia con el modelo cliente-servidor

4. DISEÑO DEL SISTEMA

Para realizar el diseño del sistema *VPS* se ha procedido según los bloques que lo conforman, es decir, se han diseñado por separado la parte correspondiente al cliente y la parte correspondiente al servidor. En lo correspondiente al servidor, se ha diferenciado la funcionalidad correspondiente al módulo de reconocimiento de imágenes de la parte de presentación de resultados al usuario.

Según el tipo de consulta que se realice en el sistema *VPS*, se pueden diferenciar dos casos de uso que son los mostrados en la figura 12 y que determinan en gran medida el diseño de la aplicación.

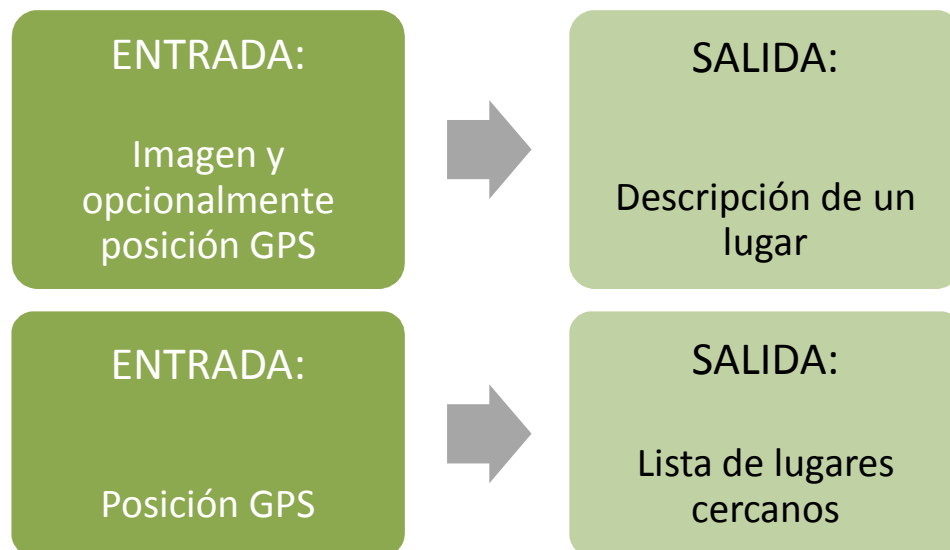


Figura 12. Posibles usos de la aplicación *VPS*

A continuación se realiza una descripción exhaustiva del diseño de cada una de las partes implicadas:

➤ DISEÑO DE LA INTERFAZ DE USUARIO

Con el fin de poder realizar pruebas con la aplicación diseñada, se ha diseñado una página web desde la que se accede al sistema *VPS*, aunque esta interfaz podría haber

sido sustituida por otra diferente y la aplicación seguiría manteniendo su funcionamiento característico.

La interfaz de usuario mencionada ha sido diseñada empleando lenguaje *HTML* (*HyperText Markup Language*), y es la interfaz a través de la cual el usuario realiza las peticiones. Para ello debe permitirle seleccionar el tipo de consulta deseada así como introducir los datos correspondientes. Igualmente, los resultados de la petición son también mostrados a través de la página.

Además de la página web de inicio a través de la cual se accede al sistema *VPS*, existen otras dos a las que el usuario accede en función del tipo de consulta a realizar. En la figura 13 se muestra el aspecto de dichas páginas web.

VPS (VISUAL POSITIONING SYSTEM)

Sistema de Localización Basado en GPS y Reconocimiento Visual de Imágenes



Elije el tipo de consulta que deseas realizar:

Introducir imagen y/o posición GPS para obtener descripción del monumento

Introducir posición GPS para obtener lista de monumentos situados a cierta distancia máxima

A continuación puedes obtener la descripción del monumento del que has tomado la foto

Indica la imagen

Indica la posición GPS (opcional):

Latitud (Grados, Minutos, Segundos)

☐ N ☐ S

Longitud (Grados, Minutos, Segundos)

☐ W ☐ E

A continuación puedes obtener una lista de sitios cercanos a la posición actual.

Indica la posición GPS:

Latitud (Grados, Minutos, Segundos)

☐ N ☐ S

Longitud (Grados, Minutos, Segundos)

☐ W ☐ E

Indica la distancia máxima de búsqueda (en metros)

Figura 13. Páginas de inicio y de consulta de una descripción y de sitios cercanos

Como se puede apreciar en la imagen anterior, la página de inicio permite básicamente decidir si se desea obtener la descripción de una ubicación o una lista de sitios cercanos a visitar.

En la página de consulta de la descripción, se solicita al usuario que introduzca la fotografía del lugar cuya descripción se desea obtener, así como su posición *GPS* (con latitud y longitud en formato de grados, minutos, segundos y orientación) si lo considera oportuno.

Análogamente, en la página de solicitud de la lista de sitios próximos, se solicita al usuario que introduzca su localización *GPS* con el formato mencionado anteriormente.

Tanto la descripción del monumento, como la relación de lugares cercanos, son visualizadas por el usuario a través de una página web, de la cual se muestra un ejemplo en la figura 14.


INICIO DE LA PETICIÓN: 00:59:23

Se ha introducido una posición GPS. Se realizará la consulta en base a la posición y la imagen introducidas.

ESTA ES LA INFORMACIÓN SOBRE EL SITIO CONSULTADO:

La Plaza de la Villa está situada en el casco histórico de Madrid (España), junto a la calle Mayor, que conforma su cara septentrional. En ella tienen su origen tres pequeñas calles, correspondientes al primitivo trazado medieval de la ciudad: la del Codo aparece por el este, la del Cordón por el sur y la de Madrid por el oeste. En su contorno se encuentran las fachadas principales de tres edificios de gran valor histórico-artístico, levantados en diferentes siglos. El más antiguo es la Casa y Torre de los Lujanes (siglo XV), construido en estilo gótico-mudéjar, que se emplaza en la cara oriental de la plaza. Le siguen en antigüedad la Casa de Cisneros (siglo XVI), un palacio plateresco que cierra la parte meridional del recinto, y la Casa de la Villa (siglo XVII), de estilo barroco, una de las sedes del Ayuntamiento de Madrid, ubicada en la zona occidental de la plaza. La Plaza de la Villa fue uno de los principales núcleos del Madrid medieval, dada su ubicación equidistante entre la puerta de Guadalajara y la de la Vega, dos de los accesos más importantes de la ciudad durante la Edad Media. Antiguamente era conocida como Plaza de San Salvador, por la iglesia del mismo nombre que se alzaba en la calle Mayor, en cuyo pórtico se realizaban las sesiones del Ayuntamiento, y que en la actualidad es recordada por una lápida que se encuentra en el sitio donde se levantaba el templo (aproximadamente en el número 70 de la citada vía). En el siglo XV, la Plaza de la Villa adoptó su actual denominación, coincidiendo con la otorgación del título de Noble y Leal Villa recibido por Madrid, de manos del rey Enrique IV de Castilla (1425-1474).

FIN DE LA PETICIÓN: 00:59:32



Ir a la página de inicio y realizar una nueva consulta

Nueva consulta

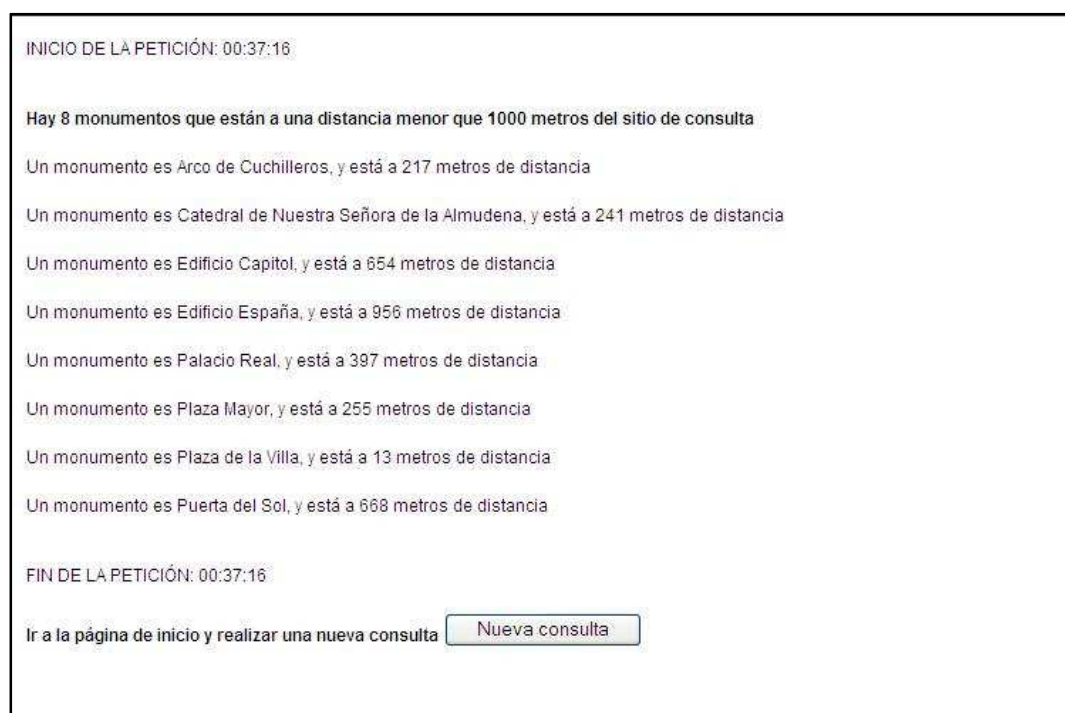


Figura 14. Páginas con resultados de consulta de una descripción y de sitios cercanos

Como bien se ha mencionado anteriormente, la implementación de la interfaz de usuario a través de un cliente web es sólo uno de los múltiples diseños posibles, ya que se podría haber utilizado una tecnología diferente para poder interactuar con la aplicación VPS.

➤ INTERACCIÓN CON EL CLIENTE

La página web desde la que se accede al sistema VPS permite introducir los datos relacionados con la búsqueda en un formulario, de modo que sean remitidos al servidor para su posterior procesamiento.

Dicho formulario ha sido construido mediante lenguaje *HTML*, y una vez que el usuario realiza la acción de solicitud de consulta, se envían los datos al servidor con el método *HTTP* "post".

La información transmitida es tratada por un programa *PHP* ("*PHP Hypertext Preprocessor*"), que interactúa con el módulo de reconocimiento de imágenes si es

preciso. Si no lo es, se extrae el resultado de la base de datos y se presenta al usuario a través de la página web.

➤ INTERACCIÓN CON EL MÓDULO DE RECONOCIMIENTO DE IMÁGENES

Cuando el usuario envía a través del formulario la fotografía de un monumento, esta imagen debe ser remitida a *FIRE*, que realiza el proceso de reconocimiento de la misma.

Para solicitar a *FIRE* que procese la fotografía, se utiliza un programa cliente de uso libre y cuyo código está disponible. Este programa, codificado en lenguaje *PHP*, permite enviar una imagen al servidor de *FIRE* y obtener las más parecidas de una base de datos que debe ser especificada previamente en un fichero de configuración.

Dado que el servidor de *FIRE* proporciona un listado con las imágenes más similares, se ha redirigido el resultado de la consulta a un fichero de texto para su posterior procesamiento por el servidor de la aplicación *VPS*.

➤ DISEÑO DE LA BASE DE DATOS

Como se ha comentado al inicio de este documento, la aplicación *VPS* permite el reconocimiento de imágenes de lugares (en este caso de la ciudad de Madrid). Para poder reconocer una imagen, el servidor de *FIRE* debe compararla con imágenes existentes en una base de datos en términos de sus características visuales (color, textura, etc.).

Para estudiar el sistema implementado, se han utilizado varias bases de datos con imágenes de los monumentos más significativos, y que por tanto tienen más probabilidad de ser fotografiados por el usuario.

Por tanto, la base de datos contiene tanto las imágenes como la información sobre sus características visuales, pero, además, también dispone de las descripciones textuales de los monumentos, así como de su posición *GPS* (necesaria para poder decidir si

están cerca o no de otro monumento en el caso de que el usuario solicite una lista de sitios próximos).

Esta última información está contenida en un documento *XML* (“*eXtensible Markup Language*”), donde los datos correspondientes a cada monumento están almacenados siguiendo la estructura determinada por un documento *DTD* al que está ligado el fichero *XML*. De este modo, cuando se requiere información sobre un sitio, el servidor accede a la base de datos, localiza en el fichero los datos correspondientes y realiza el procesamiento necesario.

4.1. DECISIONES DE DISEÑO

Además de decidir estructurar la aplicación *VPS* como un sistema cliente-servidor, ha sido necesario concretar la forma de implementar cada uno de los bloques de los que consta.

En lo que concierne al diseño de las páginas web de *VPS* se ha empleado lenguaje *HTML*, puesto que permite mostrar información tanto en forma de texto como de imágenes y satisface los requisitos de la aplicación. El desarrollo del sistema se completa con el uso de programas codificados en lenguaje *PHP*. El motivo de haber optado por dicho lenguaje, y no por *ASP*, *JSP* u otro, es su sencillez a la hora de procesar datos, en este caso los procedentes del formulario que rellena el usuario para hacer la petición. Además, no necesita compilación por ser un lenguaje interpretado, y tampoco requiere disponer de un software específico, pues es de acceso libre y gratuito.

En cuanto a la información almacenada en la base de datos, se pretende poder acceder a la misma rápidamente y de un modo directo. Una posibilidad sería emplear una base de datos propiamente dicha para el almacenamiento, pero se ha decidido estructurar los datos en un documento *XML* por las ventajas que ello supone.

El lenguaje *XML* da solución al problema de representar la información de una forma estructurada; para ello se descompone la información en elementos que son distinguidos mediante etiquetas, las cuales tienen connotaciones semánticas para dar

un mayor sentido a la estructura del documento. Como los datos de todos los monumentos tienen unas características comunes, es evidente que el uso de *XML* simplifica en gran medida el acceso que debe realizar el servidor a los mismos.

La figura 15 muestra un fragmento del documento *XML* empleado en la aplicación *VPS*.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE lista_de_sitios SYSTEM "estructura.dtd">

<lista_de_sitios>
  <sitio id="001">
    <nombre>Arco de Cuchilleros</nombre>
    <ciudad>Madrid</ciudad>
    <latitud grados="40" minutos="24" segundos="53.44" orientacion="N"></latitud>
    <longitud grados="3" minutos="42" segundos="29.01" orientacion="W"></longitud>
    <imagen>Arco_Cuchilleros.jpg</imagen>
    <descripcion>El arco de Cuchilleros es la más famosa de las nueve puertas de acceso de la Plaza Mayor de Madrid (España) y está situada en la esquina sur-oeste de la plaza. La considerable altura de este arco se debe al gran desnivel que existe entre la Plaza Mayor y la Cava de San Miguel. El arco de Cuchilleros es obra de Juan de Villanueva, quién tras el incendio de la Plaza Mayor en 1790, cerró completamente la plaza habilitando una serie de arcadas para su acceso. El origen de su nombre está en la calle de Cuchilleros a la que da salida, y en la que antiguamente se ubicaban los talleres del gremio de cuchilleros, pues suministraban sus artículos al gremio de carniceros ubicado dentro de la plaza. En la actualidad, tanto la Plaza Mayor, como el arco y calle de Cuchilleros, es un destacado punto turístico de la capital de España, estando ubicados en ellos numerosos restaurantes y bares típicos. Entre ellos se pueden citar el restaurante "Sobrino de Botín", que figura en el Libro Guinness de los Records por ser el restaurante más antiguo del mundo, puesto que fue fundado en 1725.
    </descripcion>
  </sitio>
  <sitio id="002">
    <nombre>Catedral de Nuestra Señora de la Almudena</nombre>
    <ciudad>Madrid</ciudad>
    <latitud grados="40" minutos="24" segundos="54.67" orientacion="N"></latitud>
    <longitud grados="3" minutos="42" segundos="48.22" orientacion="W"></longitud>
    <imagen>Catedral_Almutdenu.jpg</imagen>
    <descripcion>La catedral de Santa María La Real de La Almudena es la sede episcopal de la diócesis de Madrid (España). Se trata de un templo de 102 metros de longitud y 73 de altura con una mezcla de diferentes estilos: neoclásico en el exterior, neogótico en el interior y neorrománico en la cripta. Fue consagrada por el papa Juan Pablo II en su cuarto viaje a España, el 15 de junio de 1993. Es la primera catedral española consagrada por un Papa y la primera consagrada por Juan Pablo II fuera de Roma. La catedral fue consagrada por el papa Juan Pablo II el 15 de junio de 1993, tomando el relevo de la colegiata de San Isidro, que había sido la catedral provisional de Madrid desde 1885. Está ubicada en el centro de la ciudad. La fachada principal da a la plaza de la Armería, frente al Palacio Real de Madrid. A la puerta lateral se accede por la calle de Bailén y, a la cripta, desde la calle Mayor. A diferencia de otras catedrales, con una orientación este-oeste, la de la Almudena tiene una orientación norte-sur, fruto de su concepción como parte integrante del conjunto del Palacio Real de Madrid. Está construida en piedra de Novelda (Alicante) y granito de las canteras de Colmenar Viejo (Madrid).
    </descripcion>
  </sitio>
</lista_de_sitios>
```

Figura 15. Fragmento del documento *XML*

Cuando el servidor del sistema *VPS* realiza una búsqueda de información contenida en el documento *XML*, es necesario que los datos de cada monumento estén adecuadamente almacenados y se ajusten a un cierto formato. Dicho formato se ha

definido a partir de un documento *DTD* (*Document Type Definition*), que sirve para comprobar la validez de un documento *XML*, es decir, sirve de referencia al programa *PHP* empleado por el servidor para verificar que los elementos, atributos y entidades que lo componen sean correctos.

La estructura jerárquica que siguen los elementos del documento que contiene la información se representa en la figura 16.

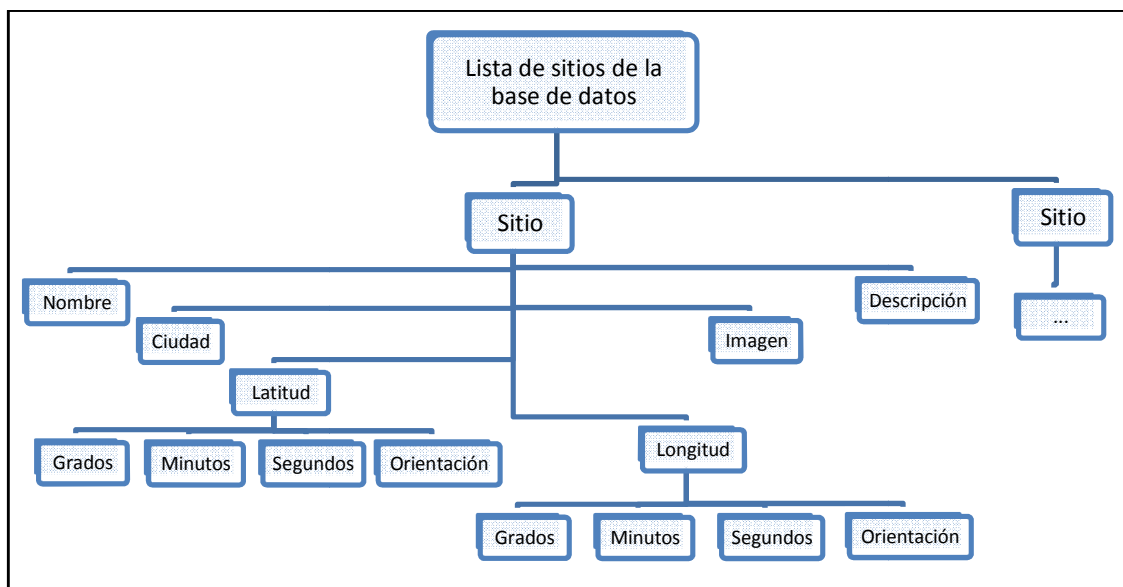


Figura 16. Estructura definida por el documento *DTD*

Según la figura anterior, se observa que la lista de monumentos recopilada en la base de datos está compuesta por sitios, cada uno de los cuales se compone a su vez de nombre, ciudad, latitud, longitud, imagen, y descripción. Asimismo, tanto la latitud como la longitud están formadas por otros elementos que son los grados, los minutos, los segundos y la orientación. Por tanto, para que la información sobre un monumento sea correcta, debe estar estructurada en el documento *XML* del modo definido en el *DTD*.

En la figura 17 se muestra el *DTD* empleado para la aplicación diseñada.

```

<!ELEMENT lista_de_sitios (sitio*)>
<!ELEMENT sitio (nombre, ciudad, latitud, longitud, imagen, descripcion)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT ciudad (#PCDATA) >
<!ELEMENT latitud EMPTY >
<!ELEMENT longitud EMPTY >
<!ELEMENT imagen (#PCDATA)>
<!ELEMENT descripcion (#PCDATA)>

<!ATTLIST sitio
  id CDATA #REQUIRED>
<!ATTLIST latitud
  grados CDATA #REQUIRED
  minutos CDATA #REQUIRED
  segundos CDATA #REQUIRED
  orientacion (N | S) #REQUIRED>
<!ATTLIST longitud
  grados CDATA #REQUIRED
  minutos CDATA #REQUIRED
  segundos CDATA #REQUIRED
  orientacion (E | W) #REQUIRED>

```

Figura 17. Estructura del documento *DTD*

Como ya se ha mencionado, en el servidor el programa *PHP* valida el documento *XML* a partir del *DTD*, y para ello se sirve de un analizador o “parser”. Existen dos mecanismos de acceso al documento *XML*: *DOM* (“*Document Object Model*”) y *SAX* (“*Simple API for XML*”). Mientras que *SAX* realiza un recorrido secuencial de los elementos del documento, *DOM* crea en memoria un árbol que contiene el documento completo de forma que luego se puede ejecutar cualquier recorrido y acción con sus elementos.

Para la implementación de la aplicación *VPS* se ha utilizado un analizador *DOM*, ya que el propio documento *XML* tiene estructura de árbol y evita el mayor trabajo de programación que es necesario con *SAX*.

Gracias al uso del analizador, será posible para el servidor leer el fichero *XML* y localizar la información que da respuesta a la petición realizada por el usuario.

4.2. BASES DE DATOS EMPLEADAS

Las bases de datos empleadas en el diseño y evaluación del sistema *VPS* contienen imágenes que corresponden a monumentos de la ciudad de Madrid. Como ya se ha mencionado, las imágenes tienen una ubicación *GPS* asociada que permite, cuando un usuario toma una fotografía, relacionarla junto con la localización *GPS* proporcionada

por el dispositivo móvil, y reconocerla dentro de la base de datos o calcular la distancia a otros monumentos. Dicha asociación entre una imagen y una posición es especificada en el fichero *XML*.

Para poder construir las distintas bases de datos manejadas, se ha utilizado la web Panoramio [24]. En este sitio web es posible acceder a fotografías de paisajes que son georreferenciadas y aportadas por los propios usuarios. Para encontrar las fotografías de las bases de datos, se ha acudido al mapa de la ciudad de Madrid y se han tomado las imágenes de los monumentos más significativos. También habría sido posible realizar una búsqueda textual de las imágenes, ya que cada una está categorizada mediante etiquetas creadas por los autores.

Con el fin de realizar una evaluación exhaustiva del sistema *VPS*, se han creado tres bases de datos cuyas propiedades se resumen en la tabla 3.

Tabla 3. Resumen de características de las bases de datos manejadas

	Nº IMÁGENES	VARIAS IMÁGENES DE UN MISMO SITIO
BASE DE DATOS "A"	23	No
BASE DE DATOS "B"	34	No
BASE DE DATOS "C"	46	Sí

La base de datos "A" es la más sencilla de las tres existentes. Contiene fotografías de 23 monumentos diferentes, pero sólo una imagen de cada uno. La base de datos "B" dispone de 34 imágenes, más que en el caso de "A" pero también con una única imagen de cada monumento. La base de datos "C" corresponde a 23 monumentos distintos, pero en este caso hay dos fotografías por cada uno de los sitios.

Estas bases de datos han sido diseñadas de modo que se pueda comprobar cómo funciona el servicio de reconocimiento de imágenes del sistema *VPS*, ya que se espera que su comportamiento sea diferente según el tamaño de la base de datos y el parecido

que guarden las imágenes entre sí (utilizando varias fotografías de un mismo sitio, habrá más imágenes que sean similares).

4.3. USO DE CBIR EN EL SISTEMA VPS

En la utilización de *VPS*, cuando un usuario realiza una fotografía, se hace necesario disponer de un servicio de reconocimiento de imágenes, tanto cuando se solicita la descripción de un monumento como una relación de sitios cercanos.

Dentro de las tecnologías *CBIR* que se han presentado en este proyecto, se ha decidido emplear *FIRE* por sus posibilidades para analizar colores, texturas y formas, y dar distinto peso a estas características ponderando los descriptores utilizados.

Cuando el sistema *VPS* interactúa con *FIRE*, le entrega una fotografía que debe ser comparada con las imágenes almacenadas en la base de datos para obtener como resultado aquellas que sean más parecidas. Esa comparación de la imagen de consulta con las almacenadas se debe realizar en términos de su contenido visual, es decir, de sus propiedades de color, textura, forma, etc.

Para ello, es necesario utilizar el servidor de *FIRE* especificando previamente la base de datos a utilizar, qué características se desean extraer, qué peso se le debe proporcionar a cada una de ellas y cuántos resultados se desean obtener. Las imágenes de la base de datos que devuelve *FIRE* como resultado se corresponden a un ranking donde están ordenadas de mayor a menor similitud, de modo que se devuelven tantas imágenes como especificadas en la configuración.

Con el fin de definir las características que deben extraerse, existen en *FIRE* unas funciones de entre las cuales se han utilizado las relacionadas con el color y la textura. Debido a que en la versión 2.2 de *FIRE*, utilizada en la implementación de *VPS*, algunas funciones presentan errores y no han sido aún correctamente implementadas, no ha sido posible extraer información relacionada con la forma.

Además de especificar las características a extraer, se debe definir cuál es la medida de comparación entre las características extraídas de cualquier conjunto de imágenes. Las distancias a partir de las cuales se mide la similitud entre imágenes incluyen la distancia euclídea, la correlación cruzada, el modelo *IDM* ("*Image Distortion Model*"), etc. y algunas de sus variantes.

Las posibles distancias que presenta *FIRE* son las enumeradas en la tabla 4.

Tabla 4. Lista de distancias disponibles en *FIRE*

DISTANCIAS DISPONIBLES
<i>bm25</i>
<i>crosscorrelation</i>
<i>euclidean</i>
<i>glfd</i>
<i>his</i>
<i>histpair</i>
<i>idm</i>
<i>imagedistortionmodel</i>
<i>jd</i>
<i>jsd</i>
<i>textfeature</i>
<i>weighteuclidean</i>

Junto con la distancia empleada, es imprescindible especificar para cada característica el peso que se le da en el reconocimiento de la imagen aportada por el usuario. Con esto, quedaría definido el funcionamiento del servidor de *FIRE* del que se hace uso en la aplicación *VPS*.

4.3.1. JUSTIFICACIÓN DE DISTANCIAS

A pesar de las múltiples distancias que se pueden utilizar para determinar la similitud entre imágenes, no todas son compatibles con las funciones de extracción de características empleadas en *FIRE*, por lo que se ha estudiado cuáles de ellas funcionan correctamente.

Con las distintas combinaciones posibles entre distancias y funciones de extracción se han hecho las pruebas necesarias para evaluar el funcionamiento del sistema *VPS*

diseñado. Debido a la escasa documentación que existe acerca de *FIRE*, ha hecho falta detectar qué distancias funcionan con cada una de las funciones mediante prueba y error, es decir, realizando consultas desde la página web implementada y viendo cuándo se generaba un error en el servidor.

La tabla 5 resume la compatibilidad entre distancias y funciones de extracción de características.

Tabla 5. Compatibilidad entre las distancias y características de *FIRE*

		FUNCIONES DE EXTRACCIÓN DE CARACTERÍSTICAS			
		color histogram	global texture feature	sparse color histogram	tamura texture feature pairs
DISTANCIA	<i>bm25</i>	x	x	x	x
	<i>crosscorrelation</i>	x	x	x	x
	<i>euclidean</i>	✓	✓	x	x
	<i>glfd</i>	x	x	x	x
	<i>his</i>	✓	✓	x	x
	<i>histpair</i>	x	x	x	✓
	<i>idm</i>	✓	✓	✓	✓
	<i>imagedistortionmodel</i>	✓	✓	✓	✓
	<i>jd</i>	✓	✓	✓	x
	<i>jsd</i>	✓	✓	✓	x
	<i>textfeature</i>	x	x	x	x
	<i>weighteuclidean</i>	✓	✓	✓	✓

Además de saber qué distancias se pueden emplear con cada función, se ha comprobado también cuál es la distancia que proporciona un mejor resultado para una función dada. Para ello se han realizado consultas en las que, fijando la función utilizada, se varía la distancia y se prueba con todas las que son compatibles.

Como criterio de decisión, se ha elegido aquella distancia que dé unos resultados tales que, si se prueba con una imagen perteneciente a la base de datos, dicha imagen esté entre los resultados devueltos y tenga el mayor grado de similitud posible. Este grado de similitud viene dado por un valor numérico proporcionado por *FIRE* y que aparece en el resultado asociado a cada imagen, y que toma valores del rango [0,1], siendo 1 el grado de máxima similitud.

De este modo, las distancias utilizadas con cada función, dentro del uso de *FIRE* que hace el sistema *VPS*, son las resumidas en la tabla 6 mostrada a continuación.

Tabla 6. Distancias empleadas con las funciones de extracción de características

FUNCIÓN DE EXTRACCIÓN DE CARACTERÍSTICAS	<i>color histogram</i>	<i>global texture feature</i>	<i>sparse color histogram</i>	<i>tamura texture feature pairs</i>
DISTANCIA	<i>his</i>	<i>euclidean</i>	<i>jd, jsd</i>	<i>histpair</i>

En el momento de realizar pruebas con las bases de datos para evaluar el comportamiento del sistema que se ha diseñado, se han utilizado las distancias y funciones especificadas anteriormente.

4.3.2. JUSTIFICACIÓN DEL USO DE "RELEVANCE FEEDBACK"

Como bien se ha comentado acerca del funcionamiento de *FIRE*, se trata de una tecnología que permite al usuario interactuar de modo que los resultados obtenidos se aproximen más a los deseados. Esta forma de interactuar, conocida como retroalimentación por relevancia (o en inglés "*Relevance Feedback*", *RF*), consiste en refinar los resultados obtenidos haciendo que el usuario indique si son relevantes, neutros o irrelevantes.

Tal y como está concebida la aplicación *VPS*, el uso de *RF* no tiene sentido, ya que supondría una doble consulta por parte del usuario (o bien tantas consultas como se consideren necesarias hasta refinar los resultados y obtener los deseados).

Además, esto supondría una ralentización del sistema, cuando lo que se persigue es que el usuario obtenga una descripción en un tiempo breve cuando realice una fotografía de un monumento. Por estos motivos, se ha decidido no utilizar esta funcionalidad de *FIRE* en el diseño del sistema *VPS*.

4.4. CÁLCULO DE LA DISTANCIA ENTRE DOS UBICACIONES

Para poder realizar los dos tipos de consulta que ofrece el sistema *VPS*, es necesario poder determinar la distancia entre dos puntos cuyas coordenadas vienen dadas por su latitud y su longitud.

En el caso de la consulta de la descripción de un lugar, el hecho de poder determinar la distancia posibilita la comparación entre la posición *GPS* proporcionada por el cliente con las posiciones asociadas a las imágenes que se encuentran en la base de datos. Así, se puede localizar cuál es la distancia menor, y que corresponderá al monumento donde es más probable que se encuentre el usuario, siendo por tanto el resultado de la petición.

Cuando se trata de la consulta de una relación de monumentos cercanos al punto donde se sitúa el usuario, se puede, a partir de la distancia, localizar en la base de datos monumentos que se alejen como máximo una cierta distancia, que ha sido previamente especificada por el usuario al realizar la consulta. De este modo, es posible que el servidor dé como resultados aquéllos monumentos más cercanos y que pueden ser de interés al usuario para su visita turística.

Habitualmente, la posición que proporciona cualquier módulo *GPS* instalado en un dispositivo móvil viene dada en función de la latitud y la longitud. El valor de éstas se expresa comúnmente en el formato “*grados:minutos:segundos*”, acompañado de la orientación, que puede ser norte o sur para la latitud y este u oeste para la longitud.

Conocidas las coordenadas correspondientes al usuario así como las asociadas a las imágenes de la base de datos, es posible determinar la distancia “*d*” entre ambas posiciones *GPS* a partir de la fórmula de Haversine [21] que se muestra en la figura 18.

$$\begin{aligned} dlong &= long1 - long2 \\ dlat &= lat1 - lat2 \\ a &= \sin^2\left(\frac{dlat}{2}\right) + \cos(lat1) * \cos(lat2) * \sin^2\left(\frac{dlong}{2}\right) \end{aligned}$$

$$c = 2 * \arcsin (\min (1, \sqrt{a}))$$

$$d = R * c$$

Figura 18. Fórmula de Haversine para el cálculo de la distancia

En la expresión anterior, donde R representa el valor del radio de la Tierra, es necesario emplear los valores decimales tanto de la latitud como de la longitud, en vez del valor que viene dado según “*grados:minutos:segundos*”. Además, para utilizar dichos valores con las funciones trigonométricas, deben ser convertidos a radianes previamente.

El resultado obtenido mediante estos cálculos viene expresado en kilómetros, pero para nuestro sistema *VPS* se utilizará su valor correspondiente en metros, ya que, cuando el usuario desee obtener una relación de lugares, pretenderá como es lógico que dichos lugares se sitúen a una distancia considerablemente pequeña (aproximadamente de centenas de metros).

4.5. DESCRIPCIÓN DE LAS CONSULTAS POSIBLES EN VPS

Una vez conocidas las decisiones de diseño tomadas en este proyecto, se puede proceder a una explicación detallada de los dos tipos de consulta que puede realizar el usuario en el sistema *VPS*.

Por un lado, es posible reconocer dentro de la base de datos una fotografía de un lugar tomada por un usuario, ayudándose de la información sobre la posición *GPS* si se dispone de ella, de forma que el usuario puede obtener una descripción textual del monumento que está visualizando. Por otro, gracias a la información sobre la posición *GPS* proporcionada por el dispositivo móvil, se puede conseguir una relación de monumentos a visitar situados como máximo a una cierta distancia del punto donde se encuentra el usuario.

Para poder comprender mejor el funcionamiento del sistema *VPS*, se detallan a continuación los procesos que intervienen en las consultas antes descritas a partir de diagramas de flujo en las figuras 19 y 21. Además, se realiza una descripción exhaustiva del comportamiento del sistema en los dos casos posibles.

➤ DESCRIPCIÓN DE UN MONUMENTO

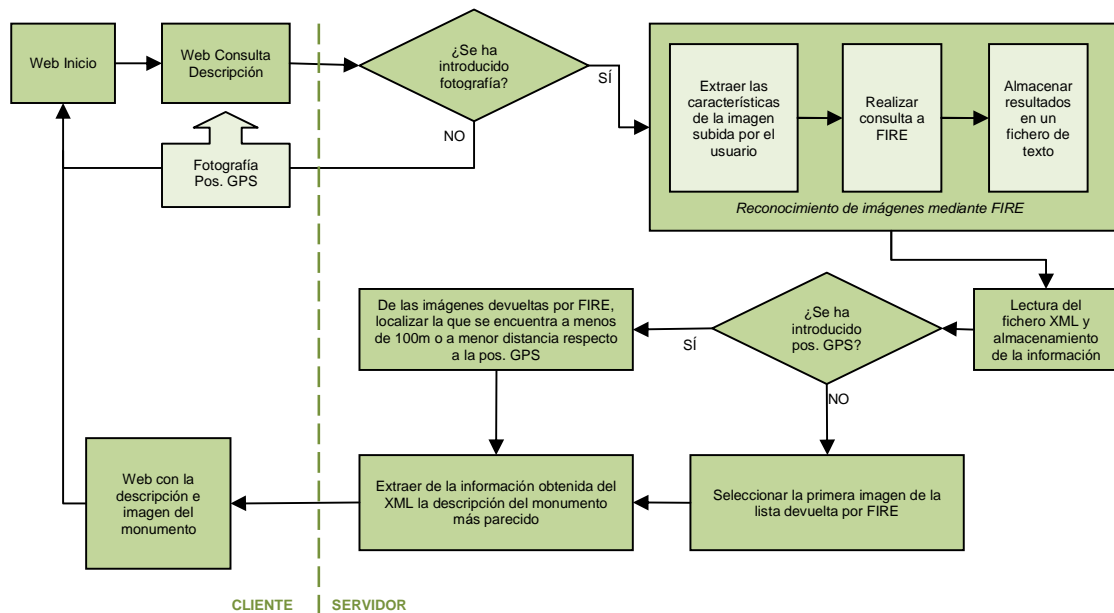


Figura 19. Diagrama de flujo para la consulta de una descripción

1. Cuando un usuario desea obtener la descripción de un monumento, debe en primer lugar visitar la página web de inicio de la aplicación VPS. A continuación, debe acceder a la página web correspondiente a la consulta de una descripción.
2. Una vez hecho esto, el usuario debe aportar los datos relativos a la consulta utilizando los campos provistos en la página con dicho fin, y que son los que se muestran en la figura 20.

Figura 20. Página de consulta de la descripción de un monumento

Para obtener la descripción, es requisito imprescindible introducir una fotografía y, opcionalmente, se pueden especificar las coordenadas *GPS*. Una vez que el usuario solicita la descripción, se envían los datos del formulario al servidor y se verifica que se haya introducido la imagen al menos. Si no es así, se redirige al usuario a la página de inicio.

3. A continuación, se realizan las tareas concernientes al reconocimiento de imágenes. Para ello, se emplean los scripts de extracción de características que se han mencionado y se realiza una petición a través del cliente de *FIRE* disponible, en la que se solicita que se reconozca la imagen aportada por el usuario. El listado de imágenes más similares devuelto por *FIRE* se almacena para su posterior procesado.

4. Puesto que toda la información de la base de datos está recopilada en el fichero *XML*, es necesario realizar la lectura de dicho archivo para luego acceder a la descripción del monumento que corresponda.

5. Posteriormente, el funcionamiento de la aplicación *VPS* difiere según se disponga de la posición *GPS* o no:

- Si no se han especificado unas coordenadas, el resultado de la consulta depende exclusivamente del reconocimiento de imágenes. Por tanto, el monumento del que se debe dar una descripción es de aquél que más se asemeja a la imagen enviada, es decir, del que figura en primera posición de los resultados devueltos por *FIRE*.
- En caso de que exista una ubicación *GPS*, se puede emplear dicha información para realizar un reconocimiento más exhaustivo de la imagen. Para ello, se debe determinar cuál es la imagen con mayor probabilidad de haber sido tomada por el usuario, es decir, la que el usuario puede estar enfocando. Como es posible que haya varios lugares próximos entre sí en la base de datos, se ha decidido discernir entre ellos de dos modos posibles: si existe algún lugar a menos de 100m (parámetro de configuración del sistema) de la posición consultada entre los proporcionados por *FIRE*, se selecciona el que ocupe una posición más alta

en el listado; en caso contrario, se toma de dicho listado el lugar que esté a la distancia mínima posible. Para llevar esto a cabo, el modo de proceder es obtener la posición de cada imagen a partir de la información obtenida del fichero *XML* y calcular la distancia respecto a cada imagen devuelta por *FIRE*. El monumento que se presenta como resultado es el que corresponde a la menor de todas las distancias calculadas.

6. Por último, se extrae la información del *XML* asociada al monumento seleccionado y se presentan tanto su descripción como su fotografía a través de la página web del sistema *VPS*. Además, se ofrece al usuario la posibilidad de realizar una nueva consulta acudiendo para ello a la página web de inicio.

➤ LISTA DE MONUMENTOS CERCANOS

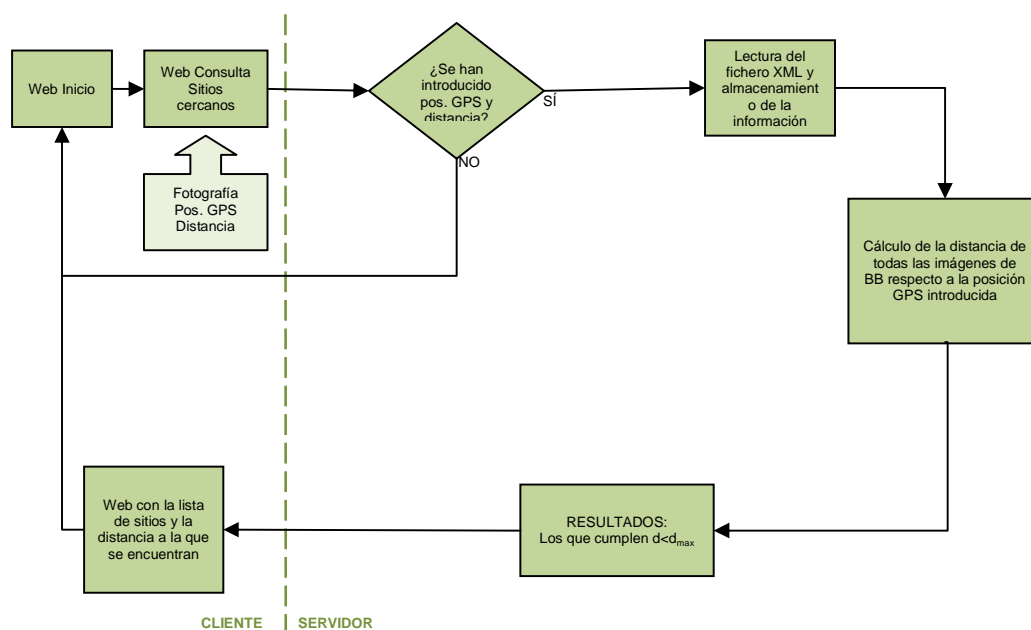


Figura 21. Diagrama de flujo para la consulta de una descripción

1. Si un usuario desea conseguir una relación de monumentos cercanos al punto donde se encuentra, debe en primer lugar acceder a la página web de inicio de la aplicación *VPS*. A continuación, debe acceder a la página web correspondiente a la consulta de una lista de sitios cercanos.

2. Una vez hecho esto, el usuario debe aportar los datos relativos a la consulta utilizando los campos provistos en la página con dicho fin, y que son los que se muestran en la figura 22.

A continuación puedes obtener una lista de sitios cercanos a la posición actual.

Indica la posición GPS:

Latitud (Grados, Minutos, Segundos) 0 0 0

☐ N ☐ S

Longitud (Grados, Minutos, Segundos) 0 0 0

☐ W ☐ E

Indica la distancia máxima de búsqueda (en metros) 200

Figura 22. Página de consulta de la descripción de un monumento

Para obtener el listado de monumentos, es requisito imprescindible introducir unas coordenadas *GPS* que sirvan como referencia para localizar monumentos cercanos al usuario. Opcionalmente, se puede aportar una fotografía que represente un monumento que está siendo visitado por el usuario en el momento de hacer la petición. Una vez que el usuario solicita la lista de sitios, se envían los datos del formulario al servidor y se verifica que se haya introducido al menos la localización. Si no es así, se redirige al usuario a la página de inicio.

3. El próximo paso consiste en realizar la lectura del archivo *XML*, ya que toda la información de la base de datos está recopilada en el mismo, y permitirá luego acceder a la información sobre la posición de los distintos monumentos y localizar los más cercanos.

4. A continuación, se debe calcular la distancia de todos los monumentos de la base de datos respecto a las coordenadas introducidas inicialmente. Como se desea obtener los sitios situados como máximo a la distancia determinada por el usuario en la

página web, se almacenarán todos aquellos monumentos situados a una distancia menor o igual que la especificada.

5. Se toman como resultados finales los que cumplen la condición de distancia especificada por el usuario.

6. Finalmente, se presentan tanto la relación de monumentos como la distancia a la que se ubican a través de la página web del sistema VPS. Además, se ofrece al usuario la posibilidad de realizar una nueva consulta accediendo para ello a la página web de inicio.

5. IMPLEMENTACIÓN

Para llevar a cabo todos los procedimientos descritos anteriormente, se ha realizado la implementación de cada elemento que interviene en el sistema VPS mediante el diseño de programas que guardan entre sí la relación descrita en el diagrama de la figura 23.

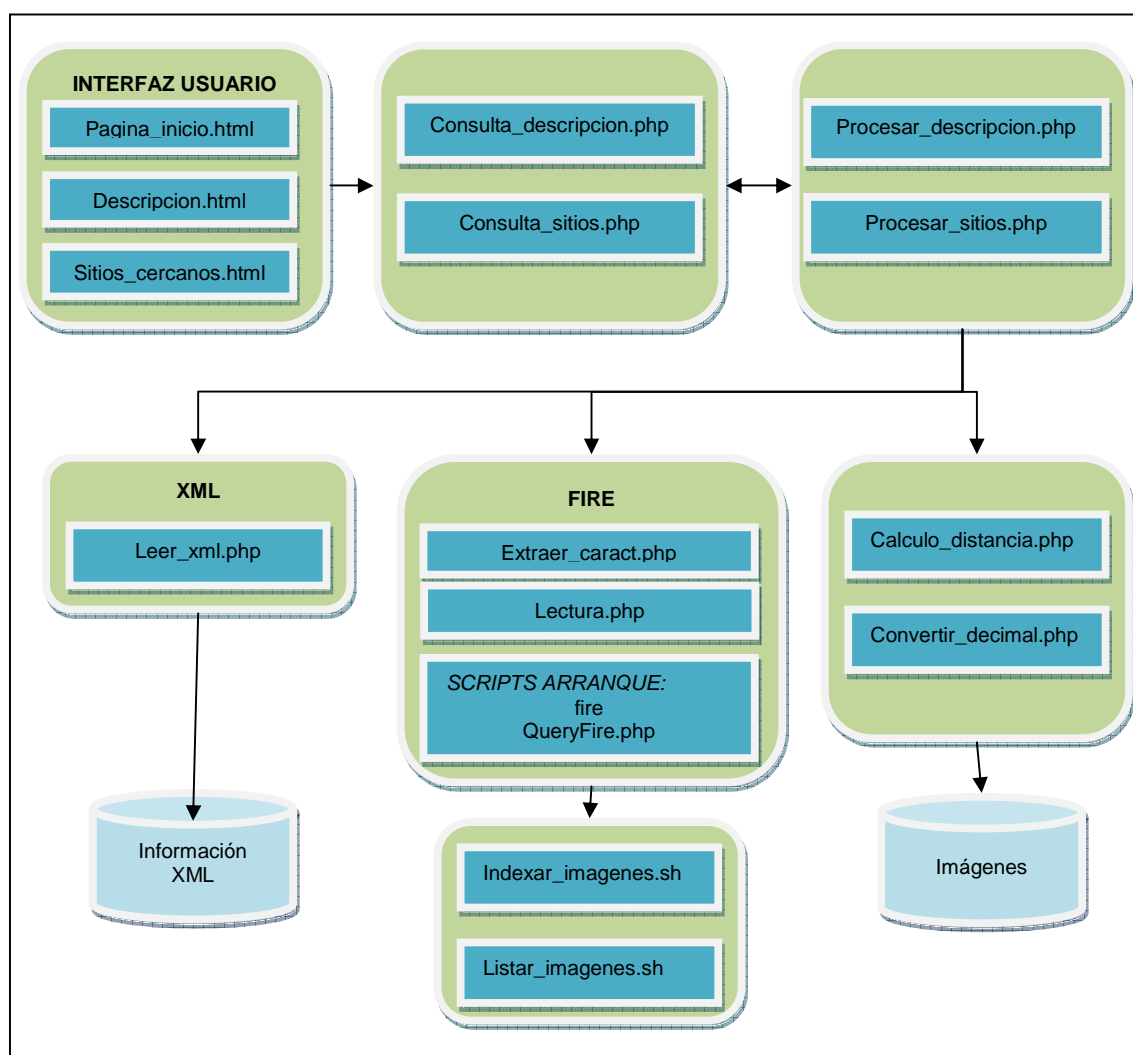


Figura 23. Relación entre los programas que intervienen en la implementación

La implementación de cada uno de los módulos necesarios se detalla a continuación.

5.1. PROGRAMAS PHP DEL SERVIDOR

Primeramente, cuando se solicita una consulta a través de la interfaz de usuario, se ejecuta uno de los scripts “*consulta_descripcion.php*” o “*consulta_sitios.php*”. Estas rutinas obtienen los datos del formulario pertinente y realizan la llamada a los scripts de procesamiento, para finalmente presentar los resultados de la consulta solicitada. En el caso de solicitar una descripción, el script presenta también como resultado la imagen de la base de datos correspondiente al monumento identificado.

Respectivamente, los scripts mencionados llaman a “*procesar_descripcion.php*” y “*procesar_sitios.php*”, que toman como entrada los datos del formulario.

Ambos scripts hacen uso de otros, también codificados en lenguaje *PHP*, y que se encargan de llevar a cabo pequeñas partes del procesamiento. Uno de estos scripts es “*extraer_caract.php*”, que realiza la extracción de características de la imagen que recibe como parámetro y genera ficheros correspondientes a las funciones de extracción de *FIRE* que se han utilizado (y que, como ya se comentó, están relacionadas con las propiedades de color y textura de la imagen). Otro programa empleado por estas rutinas es “*leer.php*”, que lee la información de salida de *FIRE* que ha sido guardada en un fichero de texto. El script “*leer_xml.php*” se encarga a su vez de la lectura del fichero *XML* donde está almacenada la información de la base de datos, y que contiene la descripción de los monumentos y su posición *GPS* asociada. Siempre que sea preciso determinar la distancia entre dos monumentos, será necesario utilizar el programa “*calculo_distancia.php*”, que determina la distancia en metros que separa dos puntos cuyas coordenadas se pasan como parámetro a partir de la fórmula de Haversine. Por último, el script “*convertir_decimal.php*” permite obtener el valor decimal de las coordenadas, ya que se deben pasar al programa “*calculo_distancia.php*” en este formato.

La rutina “*procesar_descripcion.php*” verifica primero que se haya introducido una imagen, requisito imprescindible para este tipo de consulta, y diferencia dos casos posibles según se haya introducido una posición *GPS* o no, ya que el funcionamiento de la aplicación *VPS* varía. Además, es preciso comprobar que se haya realizado correctamente la carga del archivo correspondiente a la fotografía. A continuación

realiza una llamada a la rutina “*extraer_caract.php*”, y luego ejecuta una petición con el script que actúa como cliente para *FIRE*, “*QueryFire.php*”, pasándole los parámetros pertinentes, y almacena los resultados en un fichero de texto que serán procesados mediante el script “*leer.php*”. Además, realiza la lectura del fichero XML mediante el programa “*leer_xml.php*”. En caso de haber introducido una posición *GPS*, para determinar cuál de los monumentos devueltos por *FIRE* es el más cercano y el que por tanto puede coincidir con el buscado por el usuario con mayor probabilidad, se emplean los scripts “*calculo_distancia.php*” y “*convertir_decimal.php*”.

Análogamente, la rutina “*procesar_sitios.php*” hace uso de los mismos scripts pero de modo diferente, ya que como se vio mediante los diagramas de flujo, los pasos en el procesamiento de esta consulta difieren ligeramente. Como es lógico, prescindirá de utilizar los scripts relacionados con el uso de *FIRE* en los casos en los que el usuario no haya aportado una fotografía.

5.2. SCRIPTS DE INDEXACIÓN DE FIRE

Con el fin de poder utilizar *FIRE*, es preciso realizar previamente la extracción de las características de las imágenes contenidas en la base de datos. Para ello se ha empleado el Shell-script “*indexar_imagenes.sh*” que, tomando como entrada un fichero con el listado de imágenes a procesar, realiza la extracción empleando funciones definidas en *FIRE* y genera como salida ficheros diferentes para cada imagen y función empleada. Estos archivos son luego los empleados por *FIRE* para comparar la imagen de consulta con las almacenadas y localizar las que guardan mayor similitud.

Más adelante se verán los pasos detallados para configurar *FIRE*, y se verá que es necesario emplear un fichero donde se listan las imágenes de la base de datos. Para esto, se ha utilizado otro Shell-script, “*listar_imagenes.sh*”, que lista en el fichero de configuración las imágenes existentes en un directorio que se le pasa como parámetro.

5.3. SCRIPTS DE ARRANQUE DE FIRE

Como ya se ha comentado con anterioridad, la aplicación *VPS* debe utilizar tanto el cliente como el servidor de *FIRE* siempre que sea necesario procesar una fotografía aportada por el usuario mediante reconocimiento de imágenes.

Para arrancar el servidor, se debe utilizar el script “*fire*” disponible como código abierto de *FIRE* y que recibe como parámetros el fichero de configuración a emplear, el número de resultados que debe proporcionar como respuesta, las distancias utilizadas para la comparación de las imágenes y el peso que se le da a cada característica de extracción.

Para realizar peticiones a *FIRE*, se emplea la rutina “*QueryFire.php*” que recibe como parámetros el nombre del servidor web y el puerto a través de los cuales se realiza la conexión con *FIRE*, así como la imagen que se debe reconocer y la relevancia que tiene la imagen en la búsqueda (ésta se identifica con un dígito cuyo valor es 1 ó -1, para relevancias positiva y negativa respectivamente).

5.4. PÁGINA WEB DE LA INTERFAZ

Como bien se ha dicho antes, la interfaz de usuario consiste en una página que permite acceder al servicio en el que se basa la aplicación *VPS*. Tanto la página de inicio como la correspondiente a cada consulta han sido implementadas mediante *HTML*.

La página web de inicio consta de dos formularios que redirigen a una página diferente según el botón que marque el usuario para solicitar la consulta deseada.

A su vez, las páginas a las que accede el usuario se basan también en un formulario, que en este caso arranca un programa *PHP* cuando se solicita la consulta. La rutina que debe ser ejecutada se especifica en el campo “*action*” del formulario, y será “*consulta_descripcion.php*” o “*consulta_sitios.php*” según se desee obtener la descripción de un monumento o un listado de sitios próximos, respectivamente.

En los dos casos posibles, el formulario consta de un campo “*input*” de tipo “*file*” que permite cargar una fotografía. Además, para los parámetros de latitud y longitud,

dispone de campos tipo “*select*” con listas desplegables que permiten elegir el valor numérico de los grados, minutos y segundos que definen las coordenadas. Asimismo, la orientación se selecciona mediante casillas circulares que corresponden en el formulario a campos de tipo “*radio*”.

En el caso de la consulta de monumentos cercanos, existe una lista desplegable adicional a través de la que se debe especificar la distancia máxima a la que se desean localizar los monumentos. Como en el caso de los grados, minutos o segundos, se trata de un campo tipo “*select*”.

Por último, los formularios disponen de un botón de tipo “*submit*” que arranca el programa *PHP* definido cuando el usuario acciona sobre él.

Todas las páginas web empleadas en la aplicación tienen en común que emplean codificación correspondiente al alfabeto latino, y también que hacen uso de hojas de estilo *CSS* (en inglés, “*Cascade Style Sheet*”), las cuales permiten definir la presentación y el aspecto de las páginas diseñadas.

5.5. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

Para poder utilizar la aplicación *VPS* y realizar las pruebas necesarias para evaluar el funcionamiento del sistema, es necesario llevar a cabo previamente la configuración que se detalla en el Anexo I de este documento, y que se basa en la implementación antes descrita.

6. VALIDACIÓN DEL SISTEMA

Con el fin de poder caracterizar el sistema *VPS*, se han realizado una serie de pruebas que permiten valorar distintos aspectos del funcionamiento de la aplicación. Principalmente, se pretende determinar la tasa de error del sistema diseñado, pero, por otro lado, se desea estudiar cuál es el tiempo de análisis que supone una consulta realizada por el usuario y si dicho tiempo es aceptable para él.

Como se verá más adelante, la medida de la tasa de error sólo tiene sentido en el caso de consultar una descripción, ya que cuando se solicita un listado de ubicaciones cercanas los resultados dependen únicamente de la distancia calculada, y por tanto se podría decir que el sistema es perfecto.

En el sistema implementado, se entiende la tasa de error como la relación entre el número de veces que se obtiene una descripción errónea y el número de peticiones realizadas, tal y como se muestra en la figura 24.

$$Tasa\ de\ error = \frac{Número\ de\ descripciones\ erróneas\ obtenidas}{Número\ de\ peticiones\ realizadas}$$

Figura 24. Medida de la tasa de error en el sistema *VPS*

En la validación de la aplicación *VPS*, se ha considerado que una descripción es errónea cuando no se obtiene directamente en el primer intento la descripción del lugar cuya imagen se procesa. Sin embargo, es posible que, a pesar de no lograr un acierto, la imagen se encuentre entre las seleccionadas por *FIRE*, lo que también proporciona una medida de la calidad del sistema diseñado.

Para medir el tiempo de análisis, se muestra la hora actual al inicio de la consulta y también al final, una vez que se ha llevado a cabo todo el procesamiento necesario y se ha proporcionado la respuesta al usuario, y se calcula la diferencia de tiempo entre ambos instantes.

6.1. PRUEBAS REALIZADAS

Como ya se ha comentado anteriormente, se han utilizado tres bases de datos distintas para de este modo poder estudiar las prestaciones del sistema *VPS* en diferentes situaciones.

Además, se han realizado pruebas modificando los pesos otorgados a las diferentes características extraídas. Por un lado, se han realizado consultas en las que el peso asignado a las funciones de extracción relacionadas con el contenido de color es el mismo que el correspondiente a las texturas presentes en las imágenes. Por otro, se ha fijado un peso cinco veces mayor para el color que para la textura, y viceversa, ya que se espera que el reconocimiento de imágenes proporcione resultados distintos según la importancia que se le dé a las características.

Para realizar las diferentes pruebas, se dispone de una base de datos de consulta con una imagen por cada uno de los monumentos posibles, de modo que sea posible utilizarlas para intentar reconocer cada monumento y ver cuál es el comportamiento de la aplicación *VPS*.

Para cada posible combinación de pesos y base de datos, se han hecho pruebas con los distintos tipos de consulta posibles. Esto incluye consultar la descripción de un monumento introduciendo opcionalmente unas coordenadas, y buscar una lista de lugares cercanos. En cada uno de estos cuatro casos se han empleado todas las imágenes de consulta y/o su posición asociada, y se han determinado tanto la tasa de error como el tiempo de análisis. En el caso del tiempo de análisis, se han promediado los resultados obtenidos para cada imagen o posición, con el fin de obtener el resultado medio.

En lo que respecta a la configuración del sistema, es preciso realizar una serie de cambios siempre que se pretenda utilizar una base de datos diferente (las bases de datos "A", "B" o "C" ya mencionadas) o unos pesos distintos.

- *Cambios para utilizar unos pesos diferentes*

Simplemente se deben especificar los pesos deseados cuando se arranca el servidor de *FIRE*.

- *Cambios para emplear una base de datos distinta*

En primer lugar se deben reemplazar las imágenes almacenadas en la base de datos, para realizar a continuación el indexado de las mismas mediante *FIRE*. Por último, es necesario sustituir el archivo *XML* que contiene la información sobre las imágenes, sus coordenadas y su descripción.

6.2. RESULTADOS

6.2.1. CONSULTA DE LA DESCRIPCIÓN DE UN LUGAR

➤ BASE DE DATOS “A”

La base de datos “A” contiene fotografías de 23 monumentos distintos, y una única fotografía por cada uno de ellos. Los valores medios obtenidos en las pruebas realizadas son los que aparecen en las tablas 7 y 8.

Tabla 7. Resultados con base de datos “A” para descripción de monumento

	Tiempo de análisis (s)	Tasa de error (%)
Pesos iguales	5,17	91,30
Color con mayor peso	5,17	91,30
Textura con mayor peso	5,13	91,30

Tabla 8. Resultados con base de datos “A” para descripción de monumento con posición *GPS*

	Tiempo de análisis (s)	Tasa de error (%)
Pesos iguales	5,13	56,52
Color con mayor peso	5,08	43,48
Textura con mayor peso	5,22	60,87

➤ BASE DE DATOS “B”

La base de datos “B” contiene fotografías de 34 monumentos distintos, y una única fotografía por cada uno de ellos. Los valores medios obtenidos en las pruebas realizadas son los que aparecen en las tablas 9 y 10.

Tabla 9. Resultados con base de datos “B” para descripción de monumento

	Tiempo de análisis (s)	Tasa de error (%)
Pesos iguales	5,12	85,29
Color con mayor peso	5,03	88,24
Textura con mayor peso	5,21	88,24

Tabla 10. Resultados con base de datos “B” para descripción de monumento con posición *GPS*

	Tiempo de análisis (s)	Tasa de error (%)
Pesos iguales	5,00	61,76
Color con mayor peso	8,68	55,88
Textura con mayor peso	4,94	61,76

➤ BASE DE DATOS “C”

La base de datos “C” contiene fotografías de 23 monumentos distintos, y dos fotografías diferentes por cada uno de ellos. Los valores medios obtenidos en las pruebas realizadas son los que aparecen en las tablas 11 y 12.

Tabla 11. Resultados con base de datos “C” para descripción de monumento

	Tiempo de análisis (s)	Tasa de error (%)
Pesos iguales	5,13	91,30
Color con mayor peso	8,70	91,30
Textura con mayor peso	8,78	82,61

Tabla 12. Resultados con base de datos “C” para descripción de monumento con posición GPS

	Tiempo de análisis (s)	Tasa de error (%)
Pesos iguales	9,48	52,17
Color con mayor peso	8,61	47,83
Textura con mayor peso	8,48	52,17

6.2.2. CONSULTA DE UN LISTADO DE LUGARES CERCANOS

Como bien se ha comentado antes, en el caso de consultar un listado de lugares cercanos a partir de la posición GPS, el sistema presenta un comportamiento ideal y la tasa de acierto es del 100%. Además, en dicho caso el tiempo medio de análisis es de aproximadamente 1 segundo.

El resto de resultados, correspondientes a las consultas de descripción de un lugar, se muestran a continuación distinguiéndolos según la base de datos de imágenes utilizada para las pruebas.

6.3. ANÁLISIS DE LOS RESULTADOS

Conocidos los resultados de las pruebas realizadas, es posible proceder al análisis detallado de los mismos para valorar el funcionamiento de la aplicación *VPS*. Estudiando los resultados logrados en términos del tiempo de análisis y la tasa de error, así como de la posición ocupada por las imágenes en el ranking devuelto por *FIRE*, se pueden realizar las afirmaciones que figuran a continuación:

- 1) Para una base de datos y un tipo de consulta determinados, el tiempo de análisis medio permanece prácticamente constante. En general, para una combinación dada de pesos, dicho tiempo es en la mayoría de los casos de 5 segundos, y datos aislados que se alejen de este valor pueden ser debidos a sobrecargas puntuales en la red del cliente o del servidor.
- 2) Se ha comprobado que el tiempo medio de análisis en el caso de la consulta de monumentos cercanos es independiente de cuál sea la distancia máxima de búsqueda fijada o, en cualquier caso, que los efectos de variar dicha distancia (cuyo valor puede oscilar entre 200m y 10km) no son apreciables.
- 3) En el caso de la consulta de una descripción, el tiempo medio de análisis disminuye para bases de datos más pequeñas, como se muestra en la figura 25. Por tanto, el menor valor se consigue con la base de datos “A”, que es la que menor número de fotografías contiene y en la que el reconocimiento de imágenes se lleva a cabo en una base de datos de menor tamaño.

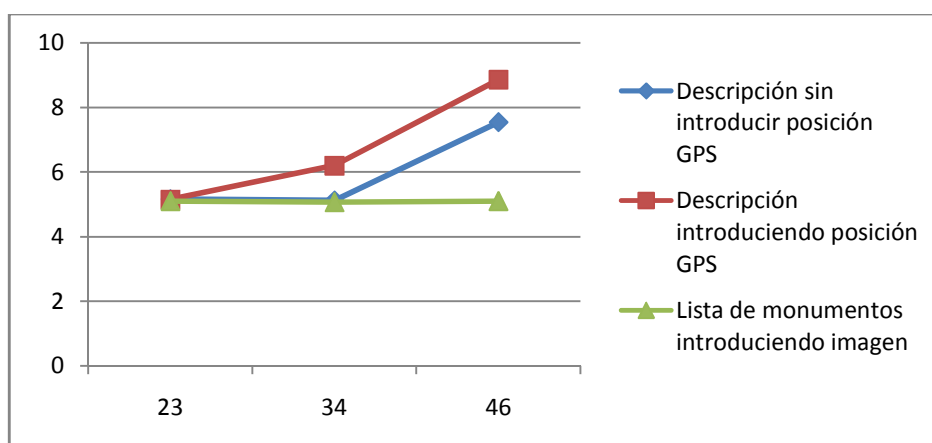


Figura 25. Relación entre tiempo de análisis y tamaño de la base de datos

- 4) En el caso de consultar un listado de ubicaciones, el tiempo medio de análisis permanece estable en torno a 5 segundos independientemente de la base de datos empleada, al contrario que en la consulta de una descripción. Esto es debido a que el cálculo de distancias que interviene en este caso es un proceso más rápido que el reconocimiento de una imagen, y su efecto se puede apreciar también en la figura 25.
- 5) Aparentemente, no existe relación entre el tiempo de análisis y los pesos que se aplican en la extracción de características de *FIRE*. Esta situación se da independientemente de la consulta llevada a cabo o de la base de datos empleada.
- 6) En el caso de consultas de una descripción en las que no se ha introducido una posición *GPS*, se obtiene una tasa de error bastante elevada para las tres bases de datos disponibles (en torno al 90%). Sin embargo, una parte considerable de las consultas realizadas tiene como resultado imágenes pertenecientes al ranking de las 10 imágenes devueltas por *FIRE*, como puede apreciarse en los datos del Anexo II.
- 7) En el caso de consultas de una descripción en la que se han aportado unas coordenadas, la tasa de error desciende notablemente, alcanzando en algunos casos valores de aproximadamente 45%. Esto se debe a que, al disponer de una posición *GPS*, se puede hacer un filtrado de los resultados aportados por *FIRE* según la distancia a la que se encuentran de la posición de entrada. Consecuentemente, es más sencillo identificar el lugar que ha sido fotografiado.

7. CONCLUSIONES Y TRABAJOS FUTUROS

A lo largo del trabajo realizado en este Proyecto Fin de Carrera se ha procedido al diseño y la validación del sistema *VPS*, que se apoya en dos técnicas muy distintas entre sí como son el posicionamiento *GPS* y el reconocimiento de imágenes.

La tecnología empleada para el reconocimiento de imágenes, *FIRE*, fue seleccionada frente a otras por las posibilidades que permitía en el funcionamiento del sistema diseñado. Como se ha visto en las pruebas realizadas, gracias a las funcionalidades de esta tecnología, se ha podido modificar el comportamiento del sistema de modo que se dé más importancia al color o a la textura de las imágenes en función de la selección de unos pesos.

Esto no hubiese sido posible con ninguna de las otras tecnologías presentadas al inicio del proyecto, lo que demuestra que la elección de *FIRE* como módulo *CBIR* es adecuada y conveniente para poder validar el sistema.

Los resultados obtenidos en la evaluación del sistema indican que las tasas de error son moderadas en algunos casos pero, en cualquier caso, las desviaciones que pueda mostrar el sistema respecto al comportamiento óptimo son debidas a las imperfecciones del sistema *CBIR* utilizado. A pesar de que actualmente se están llevando a cabo multitud de investigaciones en el área, los sistemas de reconocimiento de imágenes son aún altamente susceptibles de mejora.

A pesar de todo, hay que tener en cuenta que en la medida de las tasas de error, se ha considerado el peor caso posible, pues únicamente se considera un acierto el hecho de que la imagen aparezca en la primera posición del ranking definido por *FIRE* (o que aparezca en cualquier posición del ranking pero sea seleccionada como la más similar gracias a las coordenadas aportadas por el usuario).

Por otro lado, las pruebas llevadas a cabo han demostrado el alto grado de usabilidad que tiene el sistema en el escenario concreto donde se ha aplicado, pues el tiempo de

respuesta medio ante cualquiera de las consultas posibles es perfectamente aceptable para el usuario. De hecho, en el caso de una solicitud de lugares cercanos, la respuesta del sistema *VPS* es proporcionada casi de inmediato en un tiempo imperceptible para el usuario.

7.1. ESCALABILIDAD O APLICACIÓN A OTROS DOMINIOS

Aunque en este proyecto se ha validado la aplicación *VPS* con un determinado sistema *CBIR* y unas bases de datos con fotografías de la ciudad de Madrid, estas propiedades podrían ser modificadas.

De una forma muy sencilla, la base de datos podría ser reemplazada por otra con imágenes diferentes tanto en aspecto visual como en número, y únicamente sería necesario instalar el sistema con los nuevos parámetros para que el usuario pueda realizar peticiones. Por ejemplo, podría emplearse el sistema diseñado para localizar imágenes de yacimientos arqueológicos, exposiciones al aire libre, parques recreativos, etc. donde sea aplicable para reconocer un lugar ayudándose de unas coordenadas.

Del mismo modo, se podría trabajar con otras características diferentes del sistema *FIRE* empleado, dando prioridad a otras características visuales. Sin embargo, en este momento, existen ciertas limitaciones para ello, ya que como se ha comentado este sistema es de software libre y en la versión disponible actualmente no todas sus funcionalidades están completamente operativas. Incluso, sería posible emplear un módulo *CBIR* diferente de *FIRE*, como *GIFT* o alguno de los presentados en este trabajo, y poder comparar el funcionamiento de la aplicación en distintas situaciones.

Asimismo, sería viable mejorar el código empleado para implementar el servicio sobre el que se basa la aplicación para intentar mejorar el rendimiento del buscador y perfeccionar su funcionamiento.

7.2. TRABAJOS FUTUROS

Continuando con la línea de trabajo seguida a lo largo de este proyecto, surgen una serie de posibilidades que podrían perfeccionar el sistema diseñado.

Como se ha comentado anteriormente, la interfaz de usuario implementada mediante una página web ha sido el diseño seleccionado porque permite acceder a la aplicación VPS desde cualquier navegador y, al mismo tiempo, aporta sencillez a la solución propuesta y a la validación del sistema.

Sin embargo, dicha interfaz de usuario podría ser implementada mediante tecnologías de desarrollo de aplicaciones para dispositivos móviles, de forma que el acceso al sistema desde un dispositivo móvil estuviese estandarizado. De hecho, este posible punto de mejora está siendo desarrollado actualmente como otro Proyecto Fin de Carrera.

En una línea de trabajo ligeramente distinta, podría modificarse la aplicación diseñada para que la base de datos que contiene las imágenes esté alojada en un dispositivo móvil. Hoy en día, en que los sistemas de almacenamiento han evolucionado rápidamente, se podría almacenar imágenes de diferentes lugares en tarjetas de memoria y hacer uso de las mismas dependiendo de por dónde se realice una visita turística. Así, bastaría con modificar la base de datos y se seguiría teniendo acceso a la aplicación diseñada sin necesidad de hacer ninguna modificación ni en el servidor ni en la interfaz de usuario. Sin embargo, se perderían las ventajas conseguidas al implementar la aplicación como un sistema cliente-servidor.

Si, a pesar de todo, no fuese viable utilizar la base de datos desde el dispositivo móvil, también sería posible enviar la imagen tomada mediante *MMS* ("*Multimedia Messaging System*") a un proveedor de servicios que, teniendo acceso al servidor de la aplicación VPS, podría proporcionar al usuario la respuesta a la consulta mediante otro mensaje.

REFERENCIAS

- [1] Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier. *Modern Information Retrieval*. Primera edición. Reading, Massachusetts: Addison Wesley, Mayo 1999.
- [2] Datta, Ritendra; Joshi, Dhiraj; Li, Jia; Wang, James Z. "Image Retrieval: Ideas, Influences, and Trends of the New Age" *ACM Computing Surveys*. 2008, vol. 40, nº 2, artículo 5, p. 1 – 60.
- [3] Del Bimbo, A. *A perspective view on visual information retrieval systems* Proceedings of the 1998 IEEE Workshop on Content-Based Access of Image and Video Libraries. Santa Barbara, California: Junio 1998.
- [4] Deselaers, Thomas; Keysers, Daniel; Ney, Hermann. *FIRE – Flexible Image Retrieval Engine: ImageCLEF 2004 Evaluation*. Berlin-Heidelberg: Springer, 2005.
- [5] Deselaers, Thomas; Keysers, Daniel; Ney, Hermann. "Features for Image Retrieval: A Quantitative Comparison" *Information Retrieval*. Abril 2008, vol. 11.
- [6] Deselaers, Thomas; Keysers, Daniel; Ney, Hermann. *Classification Error Rate for Quantitative Evaluation of Content-based Image Retrieval Systems* Cambridge. 17th International Conference on Proceedings of the Pattern Recognition (ICPR'04). Cambridge, Reino Unido: IEEE Computer Society. Agosto 2004.
- [7] Eakins, J.; Graham, M. "Content-based Image Retrieval" *JTAP*. Octubre 1999, Report 39.
- [8] El-Rabbany, Ahmed. *Introduction to GPS: The Global Positioning System*. Segunda edición. Artech House, 1 Octubre 2006.
- [9] Feng, D.; Siu, W. C.; Zhang, H. J. *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*. Primera edición. Berlin-Heidelberg: Springer, Abril 2003.

- [10] Föckler, Paul, et al. *PhoneGuide: Museum Guidance Supported by On-Device Object Recognition on Mobile Phones* International Conference on Mobile and Ubiquitous Computing (MUM'05). Christchurch, Nueva Zelanda: 2005.
- [11] Hanjalic, Alan; Sebe, Nicu; Chang, Edward. *Multimedia Content Analysis, Management and Retrieval: Trends and Challenges* SPIE-IS&T vol 6073. San José, California: 15 Enero 2006.
- [12] Jesus, Rui, et al. *Geographic Image Retrieval in Mobile Guides* GIR'07: Proceedings of the 4th ACM Workshop on Geographical Information Retrieval. Lisboa: 9 Noviembre 2007.
- [13] Jin, Hai, et al. *A Flexible and Extensible Framework for Web Image Retrieval System* Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW'06). IEEE Computer Society, 19-25 Febrero 2006.
- [14] Kennedy, Michael. *The Global Positioning System and GIS*. Segunda edición. CRC, Abril 2002.
- [15] Lew, Michael S., et al. *Content-Based Multimedia Information Retrieval: State of the Art and Challenges* ACM Transactions on Multimedia Computing, Communications, and Applications. Febrero 2006.
- [16] Lim, Joo-Hwee, et al. *SnapToTell: Ubiquitous Information Access from Camera Mobile & Ubiquitous Information Access* (MUIA'04) Workshop. Glasgow, Escocia: Septiembre 2004.
- [17] McNeff, J.G. "The global positioning system" *IEEE Transactions on Microwave Theory and Techniques*. Marzo 2002, vol. 50.
- [18] Pogacnik, M.; Tasic, J.F.; Dobravec, S. *Trends in personalized content retrieval and access* Joint IST Workshop on Mobile Future, 2004 and the Symposium on Trends in Communications, SympoTIC'04. Octubre 2004.

- [19] Sasaki, Misato; Noack, Christian; Yokota, Hidetoshi; Idoe, Akira. *LocationWeb: Proposal and Implementation of Location-based Web Content Search and Creation Using a Mobile Phone* IPSJ Digital Courier, Noviembre 2005.
- [20] Singhal, Amit. *Modern Information Retrieval: A Brief Overview*. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 2001, vol. 24, p. 35 – 43.
- [21] Sinnott, R. W. "Virtues of the Haversine". *Sky and Telescope*, 1984, vol. 68, no. 2, p. 159.
- [22] Smeulders, A. W. M., et al. *Content-Based Image Retrieval at the End of the Early Years* IEEE Transactions on Pattern Analysis and Machine Intelligence. Diciembre 2000, vol. 22.
- [23] Souissi, A.; Tabout, H.; Sbihi, A. "MIR system for mobile information retrieval by image querying" *IJCSNS International Journal of Computer Science and Network Security* Abril 2008, vol. 8, nº 4.
- [24] Steiniger, Stefan; Neun, Moritz; Edwardes, Alistair. *Foundations of Location Based Services. Lecture Notes on LBS v 1.0*. Zurich: 2006.
- [25] Tanenbaum, Andrew. *Sistemas Operativos Modernos*. Segunda edición. Pearson, 2003.
- [26] *The GNU Image Finding Tool* [en línea]. Disponible en Web: <<http://www.gnu.org/software/gift/>>
- [27] Yeh, Tom; Tollmar, Konrad; Darrell, Trevor. *Searching the Web with Mobile Images for Location Recognition* Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04). Cambridge, Massachusetts: IEEE Computer Society, 27 Junio – 2 Julio 2004.

ANEXO I: INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA VPS

Antes de poder realizar pruebas con la aplicación *VPS*, es imprescindible determinar cuál es la estructura del servidor del sistema. En el caso de este Proyecto Fin de Carrera, se ha hecho uso de un servidor perteneciente a la Universidad Carlos III de Madrid, donde se ha podido alojar todo el material referente al sistema implementado. Para poder instalar correctamente el sistema, el único requisito a cumplir por el servidor es que esté basado en Linux y Apache con soporte PHP 5.

La estructura de directorios y ficheros de la que se ha hecho uso es la que aparece en la figura 26.

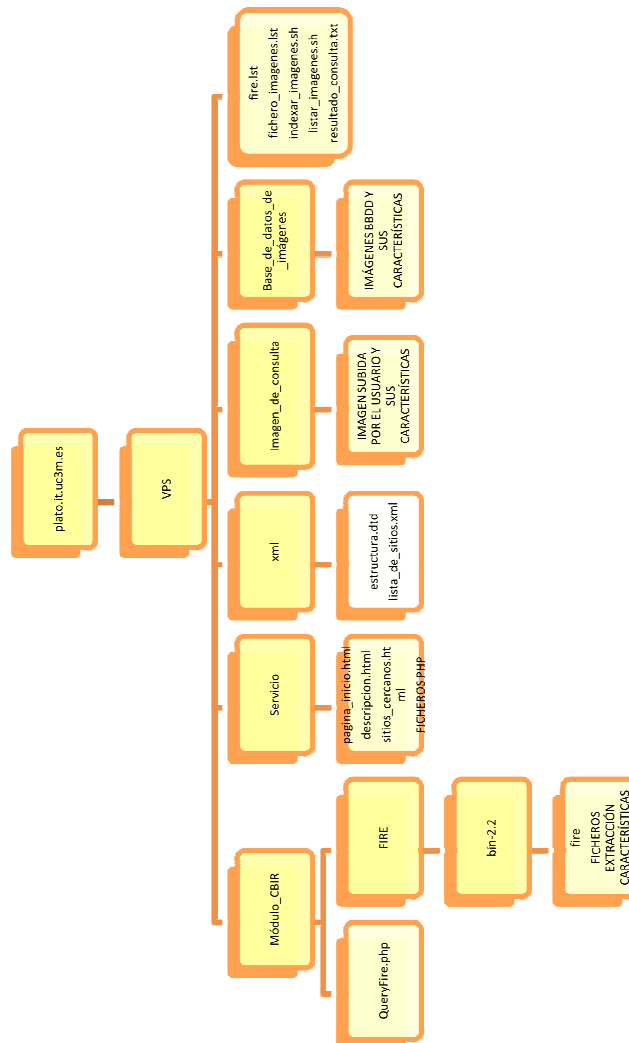


Figura 26. Estructura de directorios y ficheros del sistema *VPS* diseñado

Como se aprecia en la figura anterior, el servidor “*plato.it.uc3m.es*” aloja en el directorio denominado “*VPS*” todos los ficheros requeridos para el funcionamiento de la aplicación. En dicho directorio se ubican por separado los archivos requeridos para el empleo de *FIRE*, que se encuentran en el directorio “*Módulo_CBIR*”, y los relacionados con el resto de funciones, que están almacenados en distintas carpetas del directorio “*VPS*”.

En lo que afecta al uso de *FIRE*, es necesario emplear el archivo “*QueryFire.php*”, que actúa como cliente para realizar la petición y solicitar que reconozca una fotografía tomada por el usuario. Para lanzar el servidor de *FIRE*, se utiliza el fichero “*fire*”, junto con una serie de parámetros entre los cuales se encuentra el archivo de configuración “*fire.lst*” alojado en el directorio “*VPS*”.

Bajo el directorio “*VPS*”, hay una serie de ficheros relacionados con el uso de *FIRE*. El fichero “*fire.lst*” es un fichero de configuración en el cual se debe especificar la ruta absoluta raíz de las imágenes de la base de datos, los ficheros correspondientes a dichas imágenes y los scripts de extracción de características empleados. Los ficheros “*indexar_imagenes.sh*” y “*listar_imagenes.sh*” son Shell-scripts que permiten extraer las características de una serie de imágenes y añadir los nombres de dichas imágenes al fichero de configuración. Para dicha extracción, se emplea “*fichero_imagenes.lst*”, que contiene la lista de imágenes cuyas características se desean extraer. Por último, el archivo “*resultado_consulta.txt*” almacena la salida que da *FIRE* cuando se realiza una petición, es decir, la lista de imágenes más parecidas así como su grado de similitud.

La fotografía realizada por el usuario así como las imágenes de la base de datos, junto con sus correspondientes ficheros resultantes de la extracción de características, se ubican en los directorios “*Imagen_de_consulta*” y “*Base_de_datos_de_imágenes*” respectivamente.

Finalmente, todos los archivos que guardan alguna relación con el cliente del sistema *VPS* están almacenados en la carpeta “*Servicio*”, e incluyen tanto las páginas web como los ficheros escritos en lenguaje *PHP* que posibilitan la comunicación con el servidor de la aplicación y la de éste con el servidor de *FIRE*.

Una vez comprendida la estructura de la que hace uso la aplicación, es preciso detallar los pasos previos al uso de la misma.

En primer lugar, habiendo almacenado ya las imágenes de la base de datos en su directorio correspondiente, es necesario preparar el fichero de configuración. Para esto hace falta ejecutar los Shell-script que indexan y listan las imágenes en el fichero “*fire.lst*” que será posteriormente empleado.

El siguiente paso consiste en lanzar el servidor de *FIRE*, pues de otro modo no es posible comenzar a realizar peticiones (a no ser que, sin introducir una fotografía, el usuario desee obtener una lista de monumentos cercanos, en cuyo caso no es necesario realizar reconocimiento de imágenes mediante *FIRE*). Para ello, se debe realizar una conexión remota al servidor “plato.it.uc3m.es” y luego ejecutar el comando mostrado debajo para arrancar el servidor:

```
/home/VPS/Módulo_CBIR/FIRE/bin-2.2/fire -f fire.lst -r 10 -d 0 jsd -w 0 1
```

Con la instrucción anterior, se ejecuta el programa “*fire*” almacenado en la parte del servidor correspondiente a *FIRE* y que arranca su propio servidor para poder recibir peticiones. El comando presenta una serie de parámetros que definen el comportamiento del servidor de *FIRE*: por un lado la opción *-f* especifica el fichero de configuración, en este caso “*fire.lst*”, y la opción *-r* define el número de resultados que debe devolver *FIRE*; por otro, el resto de parámetros son relativos a las características extraídas. La opción *-d* indica la distancia a usar con una característica, *-d 0 jsd* implica que para la característica número 0 se debe usar la distancia “*jsd*”; la opción *-w* define el peso que se le da a una característica, *-w 0 1* significa que a la característica número 0 se le debe dar un peso unitario. La numeración de características, que comienza en 0, sigue el orden establecido en el fichero de configuración “*fire.lst*”.

Gracias a estas posibilidades, se puede jugar con la configuración del servidor de *FIRE* para realizar pruebas en distintas situaciones y comprobar el buen funcionamiento del sistema VPS.

Realizadas todas estas acciones, es posible comenzar a realizar consultas a través de la página web diseñada, tanto acerca de descripciones de monumentos como de listados de monumentos cercanos.

ANEXO II: RESULTADOS DETALLADOS

Las siguientes tablas recogen los datos obtenidos al realizar las pruebas con el sistema. La columna “Tiempo” representa el tiempo de análisis; la columna “Ranking” indica en el caso de reconocimiento de imágenes si la imagen de consulta es reconocida por *FIRE* como una de las 10 más similares; por último, la columna “Acierto” indica si se ha reconocido la imagen adecuadamente (en el caso de una descripción) o si la imagen cuya posición se emplea para la consulta aparece entre los resultados (en el caso de un listado de monumentos).

Para cada columna, se distinguen tres casos posibles según el peso que se asigne en *FIRE* a las distintas características: en el caso I todos los pesos son iguales, en el caso II el color tiene un peso 5 veces mayor que la textura, y en el caso III la textura tiene un peso 5 veces mayor que el color.

➤ BASE DE DATOS “A”

• DESCRIPCIÓN DE MONUMENTO SIN INTRODUCIR POSICIÓN GPS

MONUMENTO	Tiempo (s)			Ranking			Acierto		
	I	II	III	I	II	III	I	II	III
Arco de Cuchilleros	5	5	5	1	1	1	Sí	Sí	Sí
Ayuntamiento	5	6	6	3	3	2	No	No	No
Catedral Almudena	6	6	5	---	---	---	No	No	No
Cibeles	6	6	5	4	4	5	No	No	No
Congreso Diputados	5	5	5	8	5	---	No	No	No
Edificio Capitol	5	4	5	---	---	---	No	No	No
Edificio España	6	5	5	---	---	---	No	No	No
Edificio Metrópolis	5	6	5	4	7	6	No	No	No
Estación Atocha	3	3	3	---	---	---	No	No	No
Estadio S. Bernabéu	6	5	6	4	4	3	No	No	No
Faro Moncloa	5	4	5	---	---	---	No	No	No
Museo Prado	5	6	6	---	10	---	No	No	No
Neptuno	5	5	5	7	10	7	No	No	No
Palacio Real	5	5	5	2	2	4	No	No	No
Plaza Mayor	6	6	5	---	---	---	No	No	No
Plaza Toros	4	4	4	3	4	3	No	No	No
Plaza Villa	5	5	5	1	1	1	Sí	Sí	Sí
Puerta Alcalá	6	6	6	2	2	2	No	No	No
Puerta Sol	6	5	6	---	---	---	No	No	No
Templo Debod	5	6	5	---	---	---	No	No	No
Torre Picasso	5	5	5	---	7	---	No	No	No
Torres Colón	5	6	6	---	---	---	No	No	No
Torres Kío	5	5	5	---	7	---	No	No	No

- DESCRIPCIÓN DE MONUMENTO INTRODUCIENDO POSICIÓN GPS

MONUMENTO	Tiempo (s)			Ranking			Acierto		
	I	II	III	I	II	III	I	II	III
Arco de Cuchilleros	6	6	6	1	1	1	Sí	Sí	Sí
Ayuntamiento	6	5	5	3	3	2	No	No	No
Catedral Almudena	6	5	6	---	---	---	No	No	No
Cibeles	6	6	5	4	4	5	Sí	Sí	Sí
Congreso Diputados	5	5	5	8	5	---	Sí	Sí	No
Edificio Capitol	5	5	5	---	---	---	No	No	No
Edificio España	5	6	6	---	---	---	No	No	No
Edificio Metrópolis	5	5	6	4	7	6	Sí	Sí	Sí
Estación Atocha	2	3	2	---	---	---	No	No	No
Estadio S. Bernabéu	5	6	5	4	4	3	Sí	Sí	Sí
Faro Moncloa	5	4	5	---	---	---	No	No	No
Museo Prado	6	5	6	---	10	---	No	Sí	No
Neptuno	5	5	5	7	10	7	Sí	Sí	Sí
Palacio Real	5	5	5	2	2	4	Sí	Sí	Sí
Plaza Mayor	5	6	6	---	---	---	No	No	No
Plaza Toros	5	4	5	3	4	3	Sí	Sí	Sí
Plaza Villa	5	5	5	1	1	1	Sí	Sí	Sí
Puerta Alcalá	6	5	6	2	2	2	Sí	Sí	Sí
Puerta Sol	5	5	6	---	---	---	No	No	No
Templo Debod	5	6	5	---	---	---	No	No	No
Torre Picasso	5	5	5	---	7	---	No	Sí	No
Torres Colon	5	5	5	---	---	---	No	No	No
Torres Kío	5	5	5	---	7	---	No	Sí	No

- LISTA DE MONUMENTOS CERCANOS SIN INTRODUCIR FOTOGRAFÍA

MONUMENTO	Tiempo (s)	Nº sitios a d<1000m
Arco de Cuchilleros	1	8
Ayuntamiento		9
Catedral Almudena		8
Cibeles		9
Congreso Diputados		12
Edificio Capitol		11
Edificio España		8
Edificio Metrópolis		11
Estación Atocha		4
Estadio S. Bernabéu		2
Faro Moncloa		1
Museo Prado		9
Neptuno		9
Palacio Real		9
Plaza Mayor		8
Plaza Toros		1
Plaza Villa		8
Puerta Alcalá		8
Puerta Sol		12
Templo Debod		4
Torre Picasso		2
Torres Colon		5
Torres Kío		1

➤ **BASE DE DATOS “B”**• *DESCRIPCIÓN DE MONUMENTO SIN INTRODUCIR POSICIÓN GPS*

MONUMENTO	Tiempo (s)			Ranking			Acierto		
	I	II	III	I	II	III	I	II	III
Arco de Cuchilleros	6	5	6	1	1	1	Sí	Sí	Sí
Ayuntamiento	5	5	6	2	3	2	No	No	No
Biblioteca Nacional	5	5	5	---	---	---	No	No	No
Catedral Almudena	6	6	5	---	---	---	No	No	No
Cibeles	6	5	6	4	4	5	No	No	No
Congreso Diputados	5	4	6	---	7	---	No	No	No
Edificio Capitol	5	5	5	---	---	---	No	No	No
Edificio España	5	6	5	---	---	---	No	No	No
Edificio Metrópolis	5	6	6	5	7	8	No	No	No
Estación Atocha	2	2	5	---	---	---	No	No	No
Estadio S. Bernabéu	6	6	5	6	6	4	No	No	No
Estadio V. Calderón	5	6	6	---	---	---	No	No	No
Faro Moncloa	4	5	4	---	---	---	No	No	No
Hotel Palace	5	5	5	---	---	9	No	No	No
Hotel Ritz	5	5	5	---	6	---	No	No	No
Monumento Cervantes	5	6	6	---	---	10	No	No	No
Monumento Colon	5	5	4	1	3	1	Sí	No	Sí
Monumento Constitución	6	5	6	1	1	---	Sí	Sí	No
Museo Prado	6	5	6	---	---	---	No	No	No
Neptuno	5	5	5	9	---	9	No	No	No
Palacio Cristal	5	4	4	---	---	---	No	No	No
Palacio Deportes	6	6	5	2	3	4	No	No	No
Palacio Real	5	5	6	3	2	7	No	No	No
Pirulí	4	3	3	---	---	---	No	No	No
Plaza Mayor	6	5	7	---	---	---	No	No	No
Plaza Toros	4	4	5	3	5	4	No	No	No
Plaza Villa	5	5	5	1	1	1	Sí	Sí	Sí
Puerta Alcalá	6	5	5	2	2	2	No	No	No
Puerta Sol	6	6	6	---	---	---	No	No	No
Templo Debod	6	6	6	---	---	---	No	No	No
Torre Espacio	4	5	4	1	1	1	Sí	Sí	Sí
Torre Picasso	5	5	5	---	9	---	No	No	No
Torres Colon	5	5	5	---	---	---	No	No	No
Torres Kio	5	5	4	---	---	---	No	No	No

- DESCRIPCIÓN DE MONUMENTO INTRODUCIENDO POSICIÓN GPS

MONUMENTO	Tiempo (s)			Ranking			Acierto		
	I	II	III	I	II	III	I	II	III
Arco de Cuchilleros	6	11	5	1	1	1	Sí	Sí	Sí
Ayuntamiento	6	11	6	2	3	2	No	No	No
Biblioteca Nacional	5	10	5	---	---	---	No	No	No
Catedral Almudena	6	11	6	---	---	---	No	No	No
Cibeles	5	11	5	4	4	5	Sí	Sí	Sí
Congreso Diputados	5	8	5	---	7	---	No	Sí	No
Edificio Capitol	5	8	5	---	---	---	No	No	No
Edificio España	6	8	5	---	---	---	No	No	No
Edificio Metrópolis	5	9	5	5	7	8	Sí	Sí	Sí
Estación Atocha	2	5	2	---	---	---	No	No	No
Estadio S. Bernabéu	5	8	5	6	6	4	Sí	Sí	Sí
Estadio V. Calderón	5	9	5	---	---	---	No	No	No
Faro Moncloa	5	9	5	---	---	---	No	No	No
Hotel Palace	5	9	6	---	---	9	No	No	No
Hotel Ritz	5	9	5	---	6	---	No	Sí	No
Monumento Cervantes	5	10	5	---	---	10	No	No	Sí
Monumento Colon	5	8	5	1	3	1	Sí	Sí	Sí
Monumento Constitución	6	9	5	1	1	---	Sí	Sí	No
Museo Prado	5	9	5	---	---	---	No	No	No
Neptuno	5	9	5	9	---	9	Sí	No	Sí
Palacio Cristal	5	9	5	---	---	---	No	No	No
Palacio Deportes	5	12	6	2	3	4	Sí	Sí	Sí
Palacio Real	5	6	5	3	2	7	Sí	Sí	Sí
Pirulí	3	5	3	---	---	---	No	No	No
Plaza Mayor	5	9	5	---	---	---	No	No	No
Plaza Toros	4	6	5	3	5	4	Sí	Sí	Sí
Plaza Villa	5	6	4	1	1	1	Sí	Sí	Sí
Puerta Alcalá	5	11	6	2	2	2	Sí	Sí	Sí
Puerta Sol	6	9	5	---	---	---	No	No	No
Templo Debod	6	7	5	---	---	---	No	No	No
Torre Espacio	5	6	5	1	1	1	Sí	Sí	Sí
Torre Picasso	4	9	5	---	9	---	No	Sí	No
Torres Colon	5	9	5	---	---	---	No	No	No
Torres Kio	5	10	4	---	---	---	No	No	No

- LISTA DE MONUMENTOS CERCANOS SIN INTRODUCIR FOTOGRAFÍA

MONUMENTO	Tiempo (s)	Nº sitios a d<1000m
Arco de Cuchilleros	1	8
Ayuntamiento		13
Biblioteca Nacional		9
Catedral Almudena		9
Cibeles		13
Congreso Diputados		14
Edificio Capitol		12
Edificio España		9
Edificio Metrópolis		15
Estación Atocha		6
Estadio S. Bernabéu		2
Estadio V. Calderón		1
Faro Moncloa		1
Hotel Palace		11
Hotel Ritz		12
Monumento Cervantes		9
Monumento Colón		7
Monumento Constitución		1
Museo Prado		11
Neptuno		11
Palacio Cristal		5
Palacio Deportes		2
Palacio Real		10
Pirulí		2
Plaza Mayor		9
Plaza Toros		1
Plaza Villa		9
Puerta Alcalá		12
Puerta Sol		14
Templo Debod		5
Torre Espacio		1
Torre Picasso		3
Torres Colon		7
Torres Kio		1

➤ **BASE DE DATOS “C”**• *DESCRIPCIÓN DE MONUMENTO SIN INTRODUCIR POSICIÓN GPS*

MONUMENTO	Tiempo (s)			Ranking			Acierto		
	I	II	III	I	II	III	I	II	III
Arco de Cuchilleros	5	10	9	1	1	1	Sí	Sí	Sí
Arco de Cuchilleros 2	5	10	9	2	2	2	Sí	Sí	Sí
Ayuntamiento	5	9	10	2	3	2	No	No	No
Ayuntamiento 2	5	9	10	---	---	---	No	No	No
Catedral Almudena	5	9	9	---	---	---	No	No	No
Catedral Almudena 2	5	9	9	---	---	---	No	No	No
Cibeles	6	9	9	4	4	9	No	No	No
Cibeles 2	6	9	9	---	---	---	No	No	No
Congreso Diputados	5	9	8	---	---	---	No	No	No
Congreso Diputados 2	5	9	8	---	---	---	No	No	No
Edificio Capitol	5	8	9	---	---	---	No	No	No
Edificio Capitol 2	5	8	9	---	5	---	No	No	No
Edificio España	6	9	9	---	---	---	No	No	No
Edificio España 2	6	9	9	---	---	---	No	No	No
Edificio Metrópolis	5	9	9	4	10	6	No	No	No
Edificio Metrópolis 2	5	9	9	---	---	---	No	No	No
Estación Atocha	2	4	4	---	---	---	No	No	No
Estación Atocha 2	2	4	4	---	---	---	No	No	No
Estadio S. Bernabéu	6	9	9	6	7	7	No	No	No
Estadio S. Bernabéu 2	6	9	9	---	---	5	No	No	No
Faro Moncloa	4	8	8	---	---	---	No	No	No
Faro Moncloa 2	4	8	8	---	---	---	No	No	No
Museo Prado	5	10	10	---	---	---	No	No	No
Museo Prado 2	5	10	10	10	9	---	No	No	No
Neptuno	5	8	9	---	---	---	No	No	No
Neptuno 2	5	8	9	---	---	---	No	No	No
Palacio Real	5	8	9	3	3	---	No	No	No
Palacio Real 2	5	8	9	10	---	---	No	No	No
Plaza Mayor	6	10	10	---	---	---	No	No	No
Plaza Mayor 2	6	10	10	---	---	---	No	No	No
Plaza Toros	4	7	8	---	---	9	No	No	No
Plaza Toros 2	4	7	8	4	7	2	No	No	No
Plaza Villa	5	9	9	1	1	1	Sí	Sí	Sí
Plaza Villa 2	5	9	9	3	4	5	Sí	Sí	Sí
Puerta Alcalá	6	9	9	4	5	3	No	No	Sí
Puerta Alcalá 2	6	9	9	2	2	1	No	No	Sí
Puerta Sol	6	10	10	---	---	---	No	No	No
Puerta Sol 2	6	10	10	---	---	---	No	No	No
Templo Debod	6	10	9	---	---	---	No	No	No
Templo Debod 2	6	10	9	---	---	8	No	No	No
Torre Picasso	5	8	9	---	---	---	No	No	No
Torre Picasso 2	5	8	9	---	---	5	No	No	No
Torres Colon	6	9	9	---	---	---	No	No	No
Torres Colon 2	6	9	9	7	7	3	No	No	No
Torres Kio	5	9	8	---	---	---	No	No	Sí
Torres Kio 2	5	9	8	4	5	1	No	No	Sí

- DESCRIPCIÓN DE MONUMENTO INTRODUCIENDO POSICIÓN GPS

MONUMENTO	Tiempo (s)			Ranking			Acierto		
	I	II	III	I	II	III	I	II	III
Arco de Cuchilleros	9	10	9	1	1	1	Sí	Sí	Sí
Arco de Cuchilleros 2	9	10	9	2	2	2	Sí	Sí	Sí
Ayuntamiento	9	10	10	2	3	2	No	No	No
Ayuntamiento 2	9	10	10	---	---	---	No	No	No
Catedral Almudena	10	9	10	---	---	---	No	No	No
Catedral Almudena 2	10	9	10	---	---	---	No	No	No
Cibeles	10	9	9	4	4	9	Sí	Sí	Sí
Cibeles 2	10	9	9	---	---	---	Sí	Sí	Sí
Congreso Diputados	9	8	9	---	---	---	No	No	No
Congreso Diputados 2	9	8	9	---	---	---	No	No	No
Edificio Capitol	8	8	8	---	---	---	No	Sí	No
Edificio Capitol 2	8	8	8	---	5	---	No	Sí	No
Edificio España	10	9	10	---	---	---	No	No	No
Edificio España 2	10	9	10	---	---	---	No	No	No
Edificio Metrópolis	10	9	9	4	10	6	Sí	Sí	Sí
Edificio Metrópolis 2	10	9	9	---	---	---	Sí	Sí	Sí
Estación Atocha	4	4	4	---	---	---	No	No	No
Estación Atocha 2	4	4	4	---	---	---	No	No	No
Estadio S. Bernabéu	9	10	10	6	7	8	Sí	Sí	Sí
Estadio S. Bernabéu 2	9	10	10	---	---	6	Sí	Sí	Sí
Faro Moncloa	12	8	4	---	---	---	No	No	No
Faro Moncloa 2	12	8	4	---	---	---	No	No	No
Museo Prado	10	9	6	---	---	---	Sí	Sí	No
Museo Prado 2	10	9	6	10	9	---	Sí	Sí	No
Neptuno	13	9	9	---	---	---	No	No	No
Neptuno 2	13	9	9	---	---	---	No	No	No
Palacio Real	8	8	8	3	3	---	Sí	Sí	No
Palacio Real 2	8	8	8	10	---	---	Sí	Sí	No
Plaza Mayor	9	9	9	---	---	---	No	No	No
Plaza Mayor 2	9	9	9	---	---	---	No	No	No
Plaza Toros	7	7	7	---	---	9	Sí	Sí	Sí
Plaza Toros 2	7	7	7	4	7	2	Sí	Sí	Sí
Plaza Villa	8	8	8	1	1	1	Sí	Sí	Sí
Plaza Villa 2	8	8	8	3	4	5	Sí	Sí	Sí
Puerta Alcalá	9	9	10	4	5	3	Sí	Sí	Sí
Puerta Alcalá 2	9	9	10	2	2	1	Sí	Sí	Sí
Puerta Sol	15	10	10	---	---	---	No	No	No
Puerta Sol 2	15	10	10	---	---	---	No	No	No
Templo Debod	9	9	10	---	---	---	No	No	Sí
Templo Debod 2	9	9	10	---	---	8	No	No	Sí
Torre Picasso	8	8	9	---	---	---	No	No	Sí
Torre Picasso 2	8	8	9	---	---	5	No	No	Sí
Torres Colon	9	9	9	---	---	---	Sí	Sí	Sí
Torres Colon 2	9	9	9	7	7	3	Sí	Sí	Sí
Torres Kio	13	9	8	---	---	---	Sí	Sí	Sí
Torres Kio 2	13	9	8	4	5	1	Sí	Sí	Sí

- LISTA DE MONUMENTOS CERCANOS SIN INTRODUCIR FOTOGRAFÍA

MONUMENTO	Tiempo (s)	Nº sitios a d<1000m
Arco de Cuchilleros	1	8
Arco de Cuchilleros 2		8
Ayuntamiento		10
Ayuntamiento 2		10
Catedral Almudena		9
Catedral Almudena 2		9
Cibeles		10
Cibeles 2		10
Congreso Diputados		12
Congreso Diputados 2		12
Edificio Capitol		11
Edificio Capitol 2		11
Edificio España		8
Edificio España 2		8
Edificio Metrópolis		11
Edificio Metrópolis 2		11
Estación Atocha		4
Estación Atocha 2		4
Estadio S. Bernabéu		2
Estadio S. Bernabéu 2		2
Faro Moncloa		1
Faro Moncloa 2		1
Museo Prado		9
Museo Prado 2		9
Neptuno		10
Neptuno 2		10
Palacio Real		9
Palacio Real 2		9
Plaza Mayor		9
Plaza Mayor 2		9
Plaza Toros		1
Plaza Toros 2		1
Plaza Villa		8
Plaza Villa 2		8
Puerta Alcalá		8
Puerta Alcalá 2		8
Puerta Sol		13
Puerta Sol 2		13
Templo Debod		5
Templo Debod 2		5
Torre Picasso		2
Torre Picasso 2		2
Torres Colon		5
Torres Colon 2		5
Torres Kio		1
Torres Kio 2		1